CRANFIELD UNIVERSITY

AYAN CHATTERJEE

ROBUST HYPERSPECTRAL IMAGE RECONSTRUCTION FOR SCENE SIMULATION APPLICATIONS

SCHOOL OF DEFENCE AND SECURITY
Centre for Electronic Warfare, Information, and Cyber

Doctor of Philosophy
Academic Year: 2016 - 2020

Supervisor: Dr Peter W. T. Yuen
10 June 2020

CRANFIELD UNIVERSITY

SCHOOL OF DEFENCE AND SECURITY
Centre for Electronic Warfare, Information, and Cyber

Doctor of Philosophy

Academic Year: 2016 - 2020

AYAN CHATTERJEE

Robust Hyperspectral Image Reconstruction For Scene Simulation Applications

Supervisor: Dr Peter W. T. Yuen
10 June 2020

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Abstract

This thesis presents the development of a spectral reconstruction method for multispectral (MSI) and hyperspectral (HSI) applications through an enhanced dictionary learning and spectral unmixing methodologies. Earth observation/surveillance is largely undertaken by MSI sensing such as that given by the Landsat, WorldView, Sentinel etc, however, the practical usefulness of the MSI data set is very limited. This is mainly because of the very limited number of wave bands that can be provided by the MSI imagery. One means to remedy this major shortcoming is to extend the MSI into HSI without the need of involving expensive hardware investment. Specifically, spectral reconstruction has been one of the most critical elements in applications such as Hyperspectral scene simulation.

Hyperspectral scene simulation has been an important technique particularly for defence applications. Scene simulation creates a virtual scene such that modelling of the materials in the scene can be tailored freely to allow certain parameters of the model to be studied. In the defence sector this is the most cost-effective technique to allow the vulnerability of the soldiers/vehicles to be evaluated before they are deployed to a foreign ground. The simulation of a hyperspectral scene requires the details of materials in the scene, which is normally not available. Current state-of-the-art technology is trying to make use of the MSI satellite data, and to transform it into HSI for the hyperspectral scene simulation. One way to achieve this is through a reconstruction algorithm, commonly known as spectral reconstruction, which turns the MSI into HSI using an optimisation approach. The methodology that has been adopted in this thesis is the development of a robust dictionary learning to estimate the endmember (EM) robustly. Once the EM is found the abundance of materials in the scene can be subsequently estimated through a linear unmixing approach.

Conventional approaches to the material allocation of most Hyperspectral scene simulator has been using the Texture Material Mapper (TMM) algorithm, which allocates materials from a spectral library (a collection of pre-compiled endmember

materials) database according to the minimum spectral Euclidean distance difference to a candidate pixel of the scene. This approach has been shown (in this work) to be highly inaccurate with large scene reconstruction error.

This research attempts to use a dictionary learning technique for material allocation, solving it as an optimisation problem with the objective of: (i) to reconstruct the scene as closely as possible to the ground truth with a fraction of error as that given by the TMM method, and (ii) to learn materials which are trace ($<$1% in the scene) to enhance the target detection statistics of the reconstructed/ simulated scene. In this study, two methods have been proposed as alternatives to the conventional TMM approach for material allocation: (a) sparse coding dictionary with simultaneous orthogonal matching pursuit (SCD-SOMP) learning from a small batch of samples and (b) the k-means sparse coding dictionary (KMSCD) method which selects samples for dictionary training from each 'over-classified' (classified to $>$2-3 times the number of species (i.e. intrinsic dimension) in the scene) cluster to ensure all material species in the scene is included for the scene reconstruction. Furthermore, two approaches complementing the goals of the learned dictionary through a rapid orthogonal matching pursuit (r-OMP) which enhances the performance of the orthogonal matching pursuit algorithm; and secondly a semi-blind approximation of the irradiance of all pixels in the scene including those in the shaded regions, have been proposed in this work.

The main result of this research is the demonstration of the effectiveness of the proposed algorithms using real data set. The SCD-SOMP has been shown capable to learn both the background and trace materials even for a dictionary with small number of atoms ($\approx$10). Also, the KMSCD method is found to be the more versatile with overcomplete (non-orthogonal) dictionary capable to learn trace materials with high scene reconstruction accuracy ($<$1% L1 norm error). When the KMSCD is applied for scene simulation application, the material allocation for the scene is implemented by a fast non-negative orthogonal matching pursuit (FNNOMP) which shows $\approx$10x of accuracy improvement over that performed by the conventional TMM method. The scene simulation by the CameoSim simulator which adopts the proposed KMSCD and FNNOMP for scene reconstruction has shown $>$2x of accuracy enhancement over that simulated using the TMM method. Although this work has achieved an incremental improvement in spectral reconstruction, however, the need of dictionary training using hyperspectral data set in this thesis has been identified as one limitation which is needed to be removed for the future direction of research.

Keywords

dictionary, unmixing, hyperspectral, material allocation, CameoSim, scene simulation, target detection

Acknowledgements

I am incredibly grateful to Dr Peter W. T. Yuen for being a source of inspiration and for his support for this PhD study and related research. I am thankful to him for his constant constructive criticism and valuable suggestions, which benefited the academic quality of this thesis. He has been a constant source of inspiration and motivation for hard work.

Besides my primary supervisor, I also wish to acknowledge my indebtedness and sincere gratitude my co-supervisor Dr David James and internal review panel members Professor Mark Richardson and Dr Daniel Andre. I am also grateful to Albert Kirk, Glenn Hutchings and Colin Stroud from Lockheed Martin (UK) who have provided external support and clarifications whenever we required them for their CameoSim scene simulation package.

I am also thankful to the members of staff and researchers who have advised, collaborated, or shared an office with for the last three years. Lastly, I would like to thank Peter Godfree and Jonathan Piper from the Defence Science and Technology Laboratory (DSTL) for funding this research in part, their constant support and for providing the data required.

List of Publications

Journal Publications

MDPI Jimaging **Ayan Chatterjee** and Peter W. T. Yuen, ”Endmember Learning with K-Means through SCD Model in Hyperspectral Scene Reconstructions”, *Journal of Imaging*, vol. 5, no 11, Nov. 2019. doi:10.3390/jimaging5110085

Source code citation:

Ayan Chatterjee (2019) KMSCD Endmember Learning with FNNOMP representation for Hyperspectral Scene Simulators [Source Code]. doi: 10.24433/CO.5438265.v1

IEEE LOCS **Ayan Chatterjee** and Peter W. T. Yuen, ”Sample Selection with SOMP for Robust Basis Recovery In Sparse Coding Dictionary Learning”, *Letters of the Computer Society*, IEEE, vol. 2, no 3, Sept. 2019. doi:10.1109/LOCS.2019.2938446

Source code citation:

Ayan Chatterjee (2019) Sample Selection with SOMP for Robust Basis Recovery In Sparse Coding Dictionary Learning [Source Code]. doi:10.24433/CO.5073641.v2

MDPI MAKE Usman A. Zahidi[†], **Ayan Chatterjee**[†], and Peter W. T. Yuen, ”A Radiative Transfer Model-Based Multi-Layered Regression Learning to Estimate Shadow Map in Hyperspectral Images”, *Machine Learning and Knowledge Extraction*, MDPI, vol. 1, no. 3, Aug. 2019. doi:10.3390/make1030052

[†]These authors contributed equally to this work. (as mentioned in the publication)

Source code citation:

Ayan Chatterjee and Usman A. Zahidi (2019) A Radiative Transfer Model-Based Multi-Layered Regression Learning to Estimate Shadow Map in Hyperspectral Images [Source Code]. doi:10.24433/CO.8528977.v1

MDPI JImaging M. Ding, P. Yuen, J. Piper, P. Godfree, **A. Chatterjee**, U. Zahidi, S. Selvagumar, D. James, and M. Richardson, ”Design of a Tunable Snapshot Multispectral Imaging System through Ray Tracing Simulation,” *Journal of Imaging*, vol. 5, no. 1, Jan. 2019. doi:10.3390/jimaging5010009

Conference Submission

IEEE IGARSS **Ayan Chatterjee** and Peter W. T. Yuen, ”Rapid Estimation of Orthogonal Matching Pursuit Representation”, IEEE 2020 International Geoscience & Remote Sensing Symposium (IGARSS 2020), scheduled for July 19 – 24, 2020.

Source code citation:

Ayan Chatterjee (2020) Dictionary Learning with Rapid Orthogonal Matching Pursuit [Source Code]. doi: 10.24433/CO.6785856.v1

Contents

List of Figures

List of Tables

List of Abbreviations

ACE	Adaptive Coherence/Cosine Estimator
AUC	Area Under Curve
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
C-SCD	Classic Sparse Coding Dictionary
DL	Dictionary Learning
ED	Euclidean Distance
FCLS	Fully Constrained Least Squares
FLAASH	Fast Line-of-Sight Atmospheric Analysis of Spectral Hypercube
FNNOMP	Fast Non-negative Orthogonal Matching Pursuit
GSD	Ground Sampling Distance
HSI	Hyperspectral Image/Imaging
HU	Hyperspectral Unmixing
HYSIME	Hyperspectral Signal Identification By Minimum Error
LASSO	Least Absolute Shrinkage and Selection Operator
LMM	Linear Mixing Model
LS	Least Squares
LSE	Least Squares Error
LSMA	Linear Spectral Mixture Analysis
MD	Manhattan Distance
MSI	Multispectral Image
NMF	Non-negative Matrix Factorisation

NS^3	Normalized Spectral Similarity Score
OMP	Orthogonal Matching Pursuit
PC	Pearson Correlation
QUAC	QUick Atmospheric Correction
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RX	Reed-Xioli
SCD	Sparse Coding Dictionary
SD	Self-Dictionary
SAM	Spectral Angle Mapper
SID	Spectral Information Divergence
SOMP	Simultaneous Orthogonal Matching Pursuit
SVD	Singular Value Decomposition
TD	Target Detection

List of Symbols

→ X is a matrix

→ $|x|$ is the absolute value of x

→ $\bar{x}$ is the mean value of x

→ $||x||_p$ is ℓ_p norm which equates to $(\sum |x|^p)^{\frac{1}{p}}$

→ $||x||_p^q$ equates to $(\sum |x|^p)^{\frac{q}{p}}$

→ $||x||_0$ is ℓ_0 or pseudo-norm which equates to the number of non-zero entities in x

→ X^T is the transpose of X

→ n(x) refers to the number of elements in x

→ card(x) refers to the cardinality or the unique elements in x

List of Equations

Chapter 1

Introduction

1.1 Motivation of Research

Hyperspectral Imagery (HSI) contains detailed spatial and spectral information of the scene. In the detection of subpixel targets, where the size of a target is a fraction of a single pixel, hyperspectral imaging provides hundreds of bands of spectral information rather than primarily relying on spatial data to enhance the detection performance. Henceforth hyperspectral imaging has been deployed for geographical earth observation to enhance the discrimination between different land use/covers with more accurate mapping performances. HSI is conventionally deployed using push broom and whisk broom approaches. With a push broom scanner, the image is collected one line at a time (or along the track) as the aircraft moves forward with all of the pixels in a line being measured simultaneously. With a whisk broom scanner, the mirror moves back and forth to collect measurements from one pixel at a time (or across the track). Satellite sensors such as those prior to Landsat 8 have adopted the whisk broom approach. However, a push broom scanner receives a stronger signal than a whisk broom scanner because it looks at each pixel area for longer time intervals and therefore two main drawbacks are commonly occurred: (i) the detectors can have varying sensitivity, and (ii) if they are not perfectly calibrated, stripes can be seen in the image data [1].

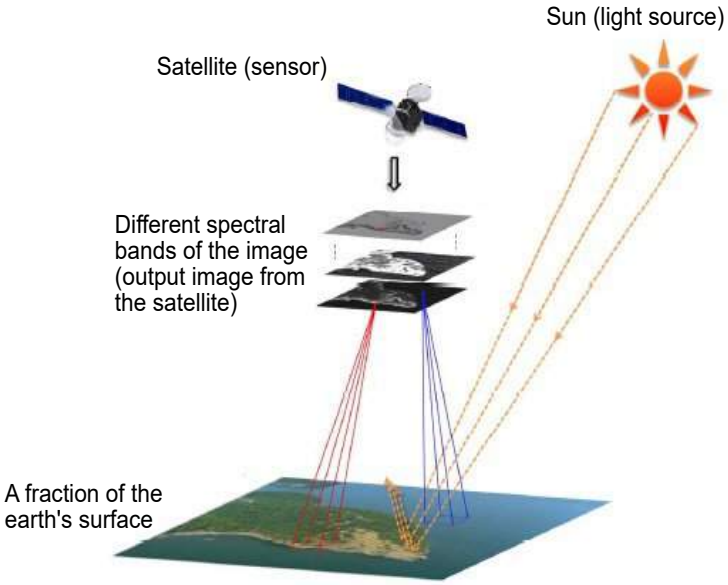
Capturing such high dimensional data comes at high manufacturing cost of the satellites particularly those with hyperspectral capabilities. Other issues such as the lack

of a common standard for manufacturing of hyperspectral sensors, and high data storage requirement where an image can be over a few gigabytes for a single frame, have been the main factor for the high cost in the HSI system. The total size of a hyperspectral image equals 'hwbB' where 'h', 'w', and 'b' are the number of rows, columns, and spectral bands of the image respectively, and 'B' is the number of bits per pixel. With such a high storage requirement, there is a huge computational cost to process the images (i.e. time consuming).

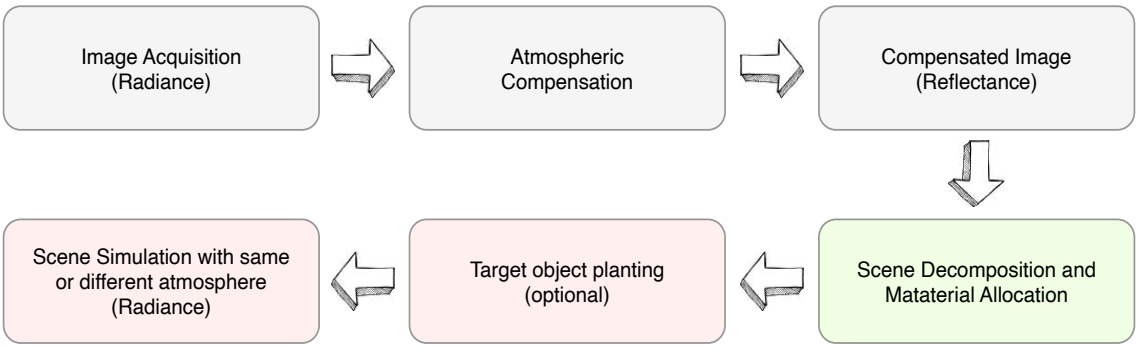
HSI has been widely used in agricultural, surveillance and reconnaissance, earth observations and military/defence applications [2, 3]. In the latter case, especially in counter-measure applications, there is a high demand for awareness of the detectability of targets when they are embedded in certain environments, such as the assessment of the detectability of diseased plants in the field or the production of sophisticated camouflage materials for different terrain and environment. In theory, this can be accomplished by repeated costly and labour intensive experimental trials until the desired result is obtained. Alternatively, this can be accomplished through effective hyperspectral scene simulation technique, which is capable of reconstructing the scene and at the same time to 'inject' or 'artificially plant' foreign materials into the environment [4, 5]. Scene simulation is an useful software tool to capture field data without the need for physical sensors.

An endmember (EM) material in a scene is a unique material type (be it a type of soil, vegetation, mineral, artificial material, etcetera) with its unique chemical composition that reflects different fractions of the incident light at different wavelengths (called spectral signature). Each pixel consists of fractional contribution of different materials, and the fractional contribution is called the abundance of material. The sum of all abundance of different materials occupying a single pixel is unity (or 100%) for a fully illuminated pixel (i.e. not in the shade). When a single material occupies an entire pixel, it is called a full pixel (or pure pixel). In practice, the approximation of full pixel materials is performed by unmixing methods.

The ultimate usefulness of a scene simulator is its ability to faithfully simulate the scene in such a way that the spectral and spatial content of the simulation is as realistic as possible. This, in turn, will require knowledge of the material property in the scene, such as the type of the material and their optical properties, to be known precisely as the input of the HSI simulator. In many cases, the detailed material properties of the scene are not known and generally only the broadband RGB or the multi-spectral image (MSI) of the scene is available [6, 7]. Spectral reconstruction (SR) is the technique to allow the 'reconstruction' of the detailed spectral characteristic of the scene from given imagery which contains only a few spectral bands. An overview of the process from acquisition to simulation is shown in figure 1.1.



(a) Image acquisition (modified from [8])



(b) Flow diagram of the process

Figure 1.1: Overview of the process from image acquisition to scene simulation

An approach to solving the inverse problem is through a 'learned' dictionary to deduce the EM characteristics of the scene through a hyperspectral unmixing (HU) algorithm [9, 10]. The learned dictionary (say a 'b×n' matrix 'D_u' with 'b' spectral dimensions/bands and 'n' EMs) is then reconstruct or rebuilt the materials of a scene through equation (1.1) where 'y' is a signal sample/pixel, 'a' is the abundance, the noise ε and the EM dictionary D_u.

$$y = D_u a + \varepsilon \tag{1.1}$$

The learned EM is then applied to the input MSI image, so that the HSI data can be reconstructed [11, 12]. This dictionary learning approach is generally conceptualised by convex optimising the linear reverse problem (LIP) with typical cost function 'J', shown in equation (1.2).

$$J = \min ||y - D_u a||_2^2 + \gamma ||a||_1 \tag{1.2}$$

The dictionary set D_u is trained from the scene, and the abundance 'a' over the dictionary exhibits characteristic high sparsity (usually $\geq$90% sparse) over the dictionary D_u; and γ the parameter for adjusting the balance between the two terms in equation (1.2). The standard approach for learning the D_u for a specific dataset is to select spectra from a comprehensive dictionary (also known as completed or overcompleted dictionary) [13], which consists of vast number of spectral database, then each of it is tested as according to equation (1.2) to justify if it fits into the criteria. The process is repeated over a comprehensive dictionary until all the D_u elements are found or until all spectral databases in the dictionary are exhausted. Previously, the comprehensive dictionary has been constructed from the pixel in the scene (called self-dictionary (SD)) [14], randomly selected from the scene [11], which sometimes in combination with spectral library data [12, 15, 16].

The quality of the reconstructed scene using the product of the learned D_u and 'a'

is critically dependent on how the D_u is learned. Furthermore, most, if not all, of the existing dictionary learning algorithms sample pixels randomly from the scene to learn. The N of pixels in the scene is in the order of millions, and over tens of thousands of training iterations (T) the computational complexity is in the order of N*T which is several orders of magnitude higher than that using KMeans clustering. Through the KMeans clustering the candidate pixels to learn is in the order of hundreds! In normal practise the random selection of test pixels from the scene never examines all the candidate pixels (ie N pixels) thus the end result exhibit large degree of non-reproducibility (ie not robustness). In the KMeans case all cluster centres are examined hence the chance to obtain the correct EM is much higher than that using random sampling method. Experimental trials include a scene with known subpixel targets to test the robustness of a reconstructed scene in terms of point to point distance difference, anomaly detection, and target detection performances.

The other issue is that in a HSI simulator's material allocation module, like in CameoSim [17] commercial military grade multispectral/hyperspectral scene simulator, the EM dictionary and the abundance is input in a lookup table (LUT) format. An additional constraint is imposed is that the LUT structure at the current stage can only accept up to a maximum of four EM materials assigned to each pixel. Therefore, one of the usefulness of this thesis is unmixing with sparsity constraint on the abundance matrix required for a simulator's material allocation module, which all the existing unmixing methods seeking true EM materials fail to realise.

The third issue is with processing complications and assumptions in an unknown scene (a captured scene with no metadata information). The main issues for the scene image that collected by air-borne sensors are : (1) the atmospheric conditions of the scene are generally not known thereby to deduce the absolute reflectance of objects on the ground cannot be obtained faithfully through compensation models like FLAASH [18], and (ii) the spectral mixing of pixels in the scene depends on the terrain and the adjacent objects (adjacency) thereby the scene in general does not conform to the linear mixing model (LMM) that assumes all pixels in the ground can be reproduced by a

linear mixture of the EM dictionary. This is particularly true when irradiance is not uniform over the terrain. Thus, in this thesis the illumination issue is tackled by the approximation of (i) the white and black calibrations to give approximate upwelling and downwelling radiance of the scene, and (ii) a binary classification map to treat the shaded regions and the non-shaded regions in two different ways.

1.2 Aims

It is evident that a faithful dictionary learning and representation algorithm is essential for hyperspectral scene simulation. Therefore, the primary aims of this thesis are:

- to realise a more robust and reproducible dictionary while decreasing the computational time of a learned dictionary,
- to increase the likelihood of learning all materials from the scene more comprehensively by using a dictionary, including minority/trace materials (materials which cover $\leq$1% abundance in training scene) which are useful for applications like anomaly detection and target detection, and
- to develop an unmixing method through dictionary learning model that seeks EM materials for a simulator's four material limit in abundance for each pixel

1.3 Introduction to hyperspectral imaging (HSI)

Remote sensing is the technique which allows the observation and monitoring of objects without physical contact with the said object. The field of study spans from celestial observation to handheld devices on the ground. Two forms of remote sensing system exist: active systems that emit radiation towards the object/scene of interest and to capture the reflected energy source such as that of the Light Detection And Ranging (LiDAR) and Synthetic Aperture Radar (SAR) technique. The second form is the passive systems which senses radiation emit from the environment when it is illuminated by natural source of energy such as the solar light source. A comparison of the two types of remote sensing is mentioned in table 1.1.

Table 1.1: The advantages and disadvantages between active and passive sources of remote sensing.

	Active	**Passive**
Advantages	better control of noise, images acquired day and night, and not affected by weather conditions	rich in contents, ability to measure temperature, and field operations are time efficient,
Disadvantages	complex field equipment, operations, and logistics, and greater survey costs	less control of noise, sensitive to weather conditions, and dependent on external energy source
Examples	LiDAR, SAR	RGB, hyperspectral

While active sources like SAR can operate at night and it is relatively unaffected by weather, it operates in the centimetre range of the electromagnetic spectrum which reduces the spatial resolution significantly in comparison to the hyperspectral imaging which operates mostly in the much shorter visible range of electromagnetic spectrum. Broadband RGB imaging is cost effective and it provides rich spatial information in three bands. Hyperspectral imaging (HSI) on the other hand provides similar spatial resolution comparable to that of the RGB, but it gives hundreds of spectral information of the scene in addition to the spatial information.

The spectral information of an object is unique to type of materials that it is made of, and in principle, a material can be distinguished from others provided that the sensing system has captured enough of spectral and spatial details. Before the launch of Hyperion Earth Observing-1 (220 bands spectral resolution) as the first space-borne imaging spectrometer in 2000, HSI spectrometers were only airborne sensors. The spatial resolution varies by their flight height, which is between 0.5 and 30 m. Since then, a range of HSI sensors have emerged:

Table 1.2: A list of commonly used hyperspectral sensors.

Name	**Spectral range**
Hyperion	0.35 to 2.57 µm
Hyperspectral Mapper (HyMap)	0.45 to 2.5 µm
Airborne Visible/Infrared Imaging Spectrometer(AVIRIS)	0.38 to 2.51 µm
HySpex	0.4 to 2.5 µm
Environmental Mapping and Analysis Program (EnMAP)	0.42 to 2.45 µm
Hyperspectral Infrared Imager (HyspIRI)	0.38 to 2.5 µm and 3 to 12 µm

HSI capture hundreds of spectral bands to make this possible. Capturing HSI images has a multitude of applications such as classification for the distinguish of one object from another, and detection of specific target from the scenes. Examples include food quality detection [19] for the identification of the freshness of meat, automatic identification of plant species [20], plant health detection through NDVI [21], for environmental research such as atmosphere monitoring and land use changes in rural/urban areas, for military applications like detection of small targets [22], etcetera. The data captured in HSI are spectral information of different materials, examples of some materials are shown in figure 1.2.

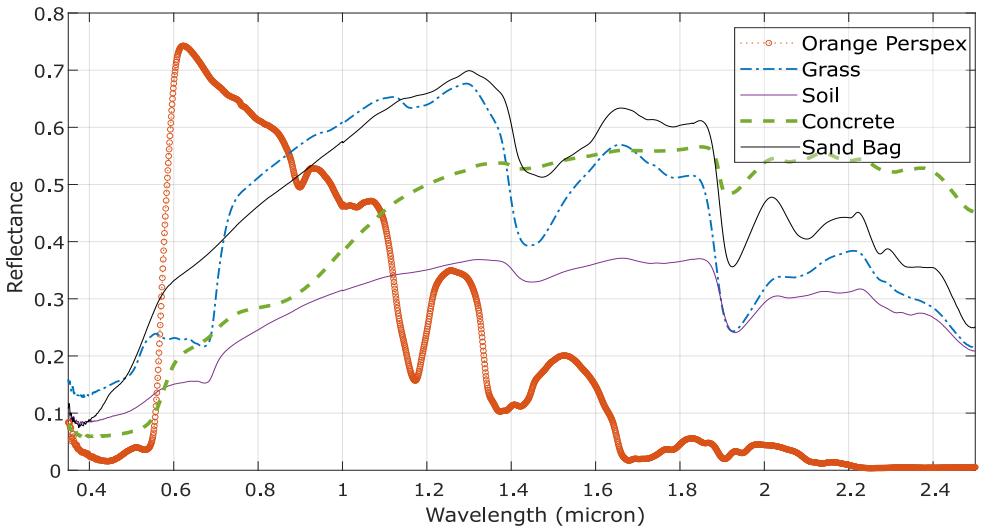


Figure 1.2: Example spectral signatures of different materials

The data captured for HSI is stored in a three-dimensional cube where the first two dimensions are the spatial dimensions of the image, and the third dimension is the spectral (or band), graphically represented in figure 1.3.

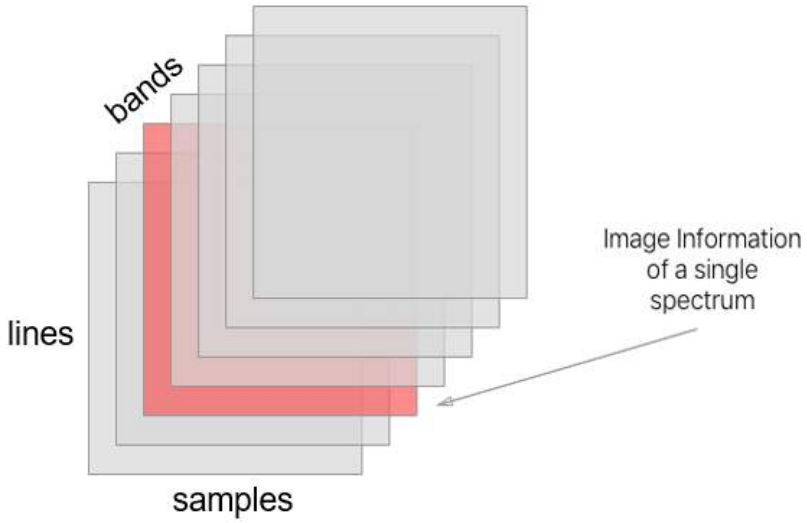


Figure 1.3: Illustration of the three dimensions in a Hyperspectral data cube.

HSI data has 4 different types of resolution in total: radiometric, spatial, spectral, and temporal. Radiometric resolution shows how accurately are the intensity details captured (example: the data represented in 2^3, 2^{10}, etcetera digits). Spatial resolution describes how big or how much of real-world detail is in a single pixel. The spectral resolution covers the wavelength range or width of a band. Lastly, temporal resolution covers how often the measurement repeats for a given target (example: 2 hours, 10 days, etcetera). By and large, every group of sensors among vast variety of available sensors is characterised by properties such as spatial, spectral, radiometric and temporal resolutions, depending on the application.

Capture a HSI resolution data comes at manufacturing cost to produce the sensors, and data acquisition and processing cost, and thus, the need to push towards dimensional reduction to capture the minimum number of bands required while maximising on the HSI applications through spectral reconstruction.

1.4 Thesis structure

The thesis is structured in the following chapters, with a flow diagram of the organisation of chapters in figure 1.4:

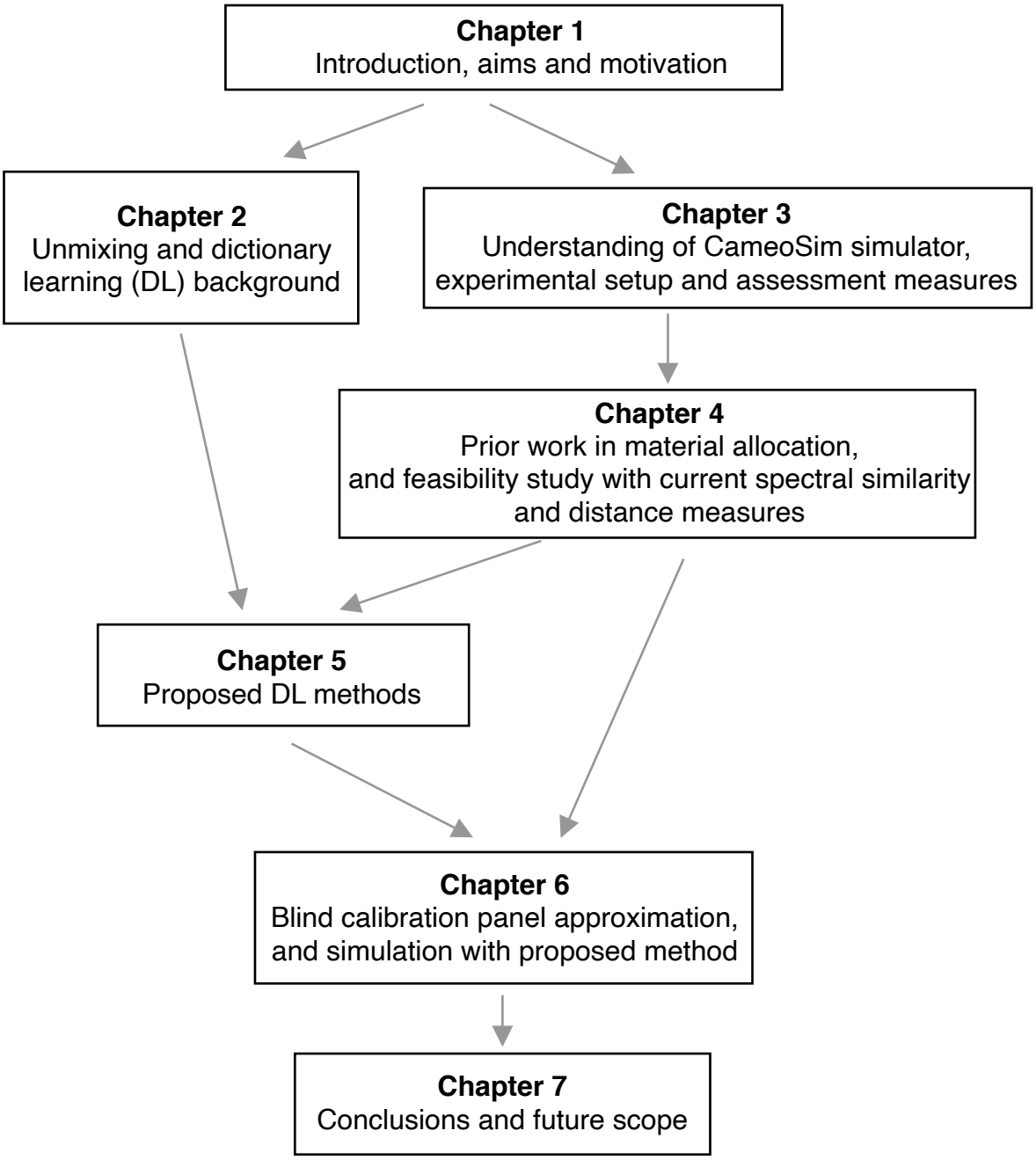


Figure 1.4: Flow diagram showing the organisation of the chapters

Introduction (Chapter 1)

This present chapter outlines the introduction to the inverse problem of scene reconstruction with dictionary learning, a brief overview into the benefits and applications of hyperspectral images, and the material allocation problem in hyperspectral simulators.

Unmixing and Dictionary Learning (Chapter 2)

This chapter provides a thorough account of the research done so far in dictionary learning (DL) and spectral unmixing (SU) methods. A pure pixel (or full pixel) is one which contains 100% of a single endmember material. In an n-dimensional data with an irregularly shaped volume (so called simplex), some pixels/data points will be inside the volume while some will be on its surface. An unmixing method will seek the corners of the simplex, and they are termed as end member (EM). Linear combination of these Ems can reproduce the pixels of the entire scene. Additionally, adjacency and scattering exist in the atmosphere, and, a full pixel can be 'contaminated' by rays scattered from other objects making it a 'mixed' pixel. SU methods traditionally estimate endmember (EM) materials, or the purest form of materials, from a simplex assuming a convex cone. SU methods in recent years are solved by adopting DL methods by solving the unmixing problem as an optimisation problem. The difference between using a convex cone and with statistical optimisation is that DL unmixing may produce unrealistic estimations of EM, specifically when there is no pure pixel in the scene, but this is the most accurate method when it comes to scene reconstructions.

CameoSim Simulator, Experimental Setup, and Assessment Metrics (Chapter 3)

This chapter provides a brief description of CameoSim hyperspectral simulator, a walk-through of the scene simulation process, and description on the structure of material allocation structure taken as input to the simulator. This chapter concludes by summarising the hyperspectral images used in this thesis, with detailed explanation on the primary dataset, and the assessment metrics used to evaluate the accuracy and robustness of both the existing and proposed models.

Material Allocation Overview (Chapter 4)

The accuracy of a material allocation method outlines the quality of the simulated imagery. This chapter outlines the prior work in material allocation. This is followed by a thorough study of material allocation with existing spectral similarity and distance measure, outlining the advantages and drawbacks of each.

Dictionary Learning Enhancements (Chapter 5)

This chapter provides the two proposed direction to enhance DL: SCD-SOMP (sparse coding dictionary with simultaneous orthogonal matching pursuit, [23]) and KMSCD (k-means sparse coding dictionary, [3]). Both SCD-SOMP and KMSCD give away with random sampling with different approaches to sampling scheme. SCD-SOMP is designed to learn trace materials for detection applications, and KMSCD is designed for faster convergence and to minimise the mean error.

Material allocation in a simulator includes a sparsity constraint of the abundance matrix. This chapter concludes by presenting KMSCD coupled with the modern fast non-negative orthogonal matching pursuit (FNNOMP) to accommodate the maximum material per limit in material allocation.

Scene simulation (Chapter 6)

This chapter presents a novel optimisation approach [24] through ELM equation to estimate white and black calibrations in a scene. By solving this through linear regression, the estimated calibrations panels can be potentially used for (i) shadow detection and masking, and (ii) approximating the atmospheric conditions of an unknown scene by systematically optimising one atmospheric parameter at a time using the approximated calibration panels. However, due to the lack of suitable ground truth data the error measurements are difficult to verify and will be part of future work. This chapter concludes by presenting CameoSim simulations with the proposed KMSCD+FNNOMP material allocation and approximated atmosphere, comparing with the existing and widely used method, texture material mapper (TMM).

Conclusions and Future Work (Chapter 7)

The ultimate usefulness of scene reconstruction in a hyperspectral simulator is with multispectral image (MSI) input from satellite imagery. This chapter summarised the proposed enhancements in this thesis. This chapter concludes with mentions of future research scope of the proposed direction, which is towards spectral reconstruction with MSI inputs in scene simulators.

Chapter 2

Unmixing and Dictionary Learning

This chapter presents the fundamental concepts of spectral unmixing and dictionary learning (DL) that are crucial to the understanding of the developments of this thesis. This is because a combination of DL and unmixing theories will be used in the proposed material allocation. Material allocation module in a scene simulator is required to input ground reflectance data of each pixel to a simulator before raytracing. Material allocation module allocates or assigns endmember material and abundance values in a tabular format while maintaining a simulator's input constraints and limitations.

The basic idea behind both unmixing and DL is to decompose a given HSI into a set of fundamental constituents that sparsely contribute to the structure of a pixel, and a non-negative condition on all decomposed components. In unmixing, the HSI decomposes into true materials, or endmembers (EM), and its abundance pair. And, in dictionary learning, the HSI decomposes into a set of basis, called atoms, and its respective representation pair.

Spectral unmixing and DL methods approach the inverse problem with a distinction between the two approaches. The distinction is that EM abundance enforces a soft ℓ_1 norm for sum-to-one, and DL atoms enforce a unit ℓ_p norm. In theory, p>0 for atomic dictionaries, however, most algorithms use ℓ_2 norm and solve the optimisation problem to minimise the Frobenius norm of the residual error.

On the EM abundance side, a soft condition in sum-to-unity is placed due the linear mixing model assumption, and the spectral differences caused by material variability and non-uniform illumination throughout a real scene. This conditions for decomposition of both DL and SU methods is illustrated in figure 2.1.

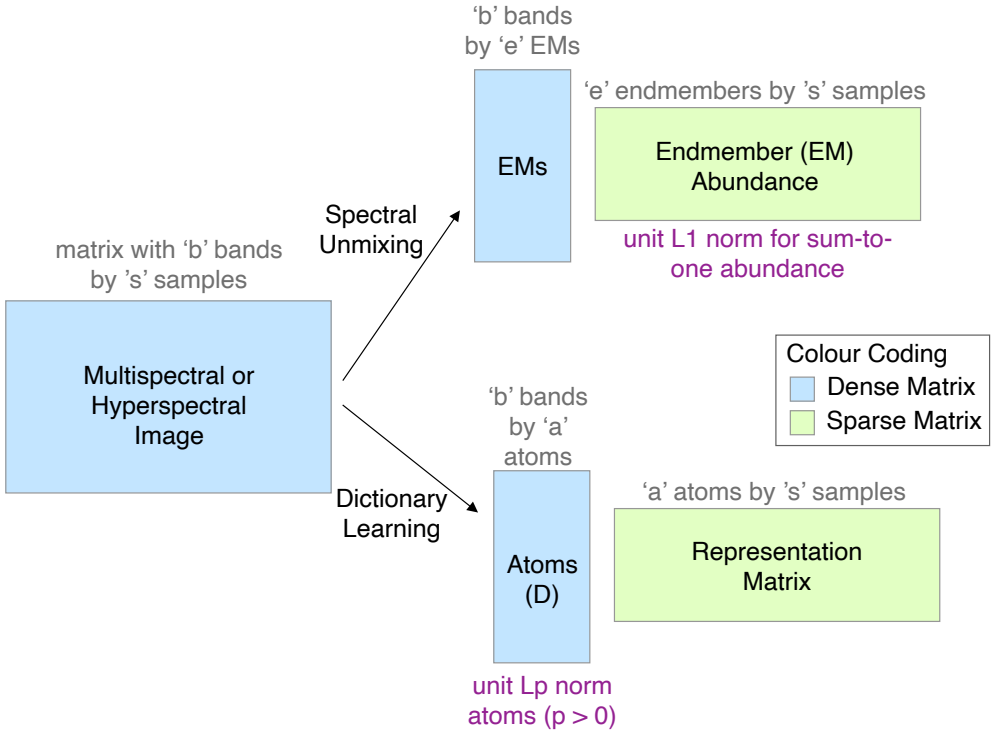


Figure 2.1: Decomposing a multi-band image into endmember-abundance pair for spectral unmixing, and, atom-representation pair for dictionary learning.

Unmixing and DL are one of many ways to split up an HSI. Other linear methods like the principal component analysis (PCA, [25, 26]) decomposing HSI to a set of linearly uncorrelated basis variables called principal components without any positive constraint on the representation, and non-linear models like the Gaussian mixture separation in [27, 28] are not feasible for a simulator's material allocation module in its present form.

With linear mixing of materials that can be potentially used in material allocation, this chapter is structured by first providing a survey of prior work with unmixing methods in section 2.1, followed by an overview and survey of DL methods in section 2.2. This study paves way for the classic sparse coding dictionary model in section 2.3, which is

the primary model used by the proposed methods in this thesis.

2.1 Spectral Unmixing (SU)

2.1.1 Principle behind unmixing

Spectral Unmixing (SU) is a method by which a collection of mixed pixels is decomposed into its true material form, or endmembers (EMs), and its corresponding fractional contribution, called abundance. EMs correspond to true form of materials in natural and artificial materials. In this thesis, linear mixture of material EM extraction is posed, where EMs are added linearly with its corresponding abundance values for a given pixel, as it is considered in simulators. Two types of EM extraction describe linear models: linear spectral mixture analysis (LSMA, [29]) describing 1 EM for each material type, and multiple EM spectral mixture analysis (MESMA, [30]) describing >1 EM for each material type observing slight variations in EM spectra a given material [31]. The general SU equation for a signal 'y' and EM dictionary 'D_u' is given by:

$$y = D_u a, \text{ where } ||a||_1 = 1 \tag{2.1}$$

The ℓ_2^2 norm of a vector 'x' is '$x^T x$', which is relatively easier to compute as compared with other norms requiring an elementwise operation, thus, by extension use of the Frobenius norm is preferred. The Frobenius norm for a matrix 'X' is mathematically represented as:

$$||X||_F = (\sum_i \sum_j x_{i,j}^2)^{0.5}, \; x_{i,j} \in X \tag{2.2}$$

The trace of a matrix is the sum of all diagonal entities, and square of the Frobenius norm is written as (where Tr refers to the trace):

$$||X||_F^2 = Tr(X^T X) \tag{2.3}$$

2.1.2 Types of linear mixtures

Different approaches to unmixing use a combination of the type of unmixing algorithm as well as constrain on the abundance matrix. In a linear mixture model (LMM), we assume that every pixel signature is a linear combination of the fractional abundance of EM materials. Three common versions of LMM describe the LMM, each with its own set of constraints on the abundance. For the purposes of nomenclature to reference the versions of LMM, the methods will be referred to as LMM T1 (or type 1), T2, and T3 respectively. The three types of LMM and the approaches to solve them is mentioned in Chang's paper in [32] and Greer's paper in [33]. The types are:

LMM T1: Fully constrained least squares (FCLS) abundance

FCLS is the common understanding of LMM, where both abundance non-negative and sum-to-one abundance is considered. Mathematically, it is written as:

$$Y = D_u a,\ \forall a_i \geq 0 \text{ and } \sum_i a_i = 1 \tag{2.4}$$

LMM T1 in its proper form is computationally expensive, with the need to estimate two Lagrange multipliers: one for the non-negative condition, and the second for sum-to-unity. A common workaround to estimate T1 abundance is to include only one Lagrange multiplier for the non-negative condition which is solved by Karush–Kuhn–Tucker (KKT) conditions, and the sum-to-one is met by adding a ones vector to the EM dictionary and the input image signals. The modified EM dictionary and input signals with the initial 'b' will now contain 'b+1' bands. The updated EM dictionary matrix is:

$$\begin{pmatrix} em_1 & em_2 & em_3 & \dots & em_e \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix}$$

LMM T2: Abundance with room for variation in illumination

The second version of LMM is where abundance is both non-negative and upper bounded sum-to-unity. This is the more realistic to consider than T1 as real scenes do have illumination artefacts. The unit sum of the abundance is considered for illumination or shade, where $\sum_i a_i = 1$ is iff the given pixel is fully illuminated. It is mathematically written as:

$$Y = D_u a,\ \forall a_i \geq 0 \text{ and } \sum_i a_i \leq 1 \tag{2.5}$$

A common approach to convert LMM T2 to T1 is by adding a zeros vector to the EM dictionary, where the modified EM dictionary with a maximum of 'e' EMs will now contain 'e+1' EMs. The abundance estimated with this modified EM dictionary will equal sum-to-unity. The updated EM dictionary matrix is:

$$\begin{pmatrix} em_1 & em_2 & em_3 & \dots & em_e & em_{e+1} \end{pmatrix}$$

where all the individual elements of em_{e+1} is 0.

LMM T3: Abundance with only non-negative constraint

The third version of LMM only considers the non-negative condition of EM abundance, with no upper bound on the abundance sum. It is mathematically represented as:

$$Y = D_u a,\ \forall a_i \geq 0 \tag{2.6}$$

An approach to convert LMM T3 to a T2 is a two simple step process, where the first step is to estimate the maximum sum of abundance present (say 'm_a'). The second step is to divide the abundance of all samples/pixels by the maximum 'm_a' and multiplying EM dictionary by 'm_a'. Material allocation in scene simulators follow a strict T1 type of LMM, however, T2 and T3 methods can be incorporated with the help of these conversion approaches.

2.1.3 Prior work in Spectral Unmixing

There are two main approaches towards the solution of unmixing equation previously mentioned in (2.1): one is to find the purest element in the scene through searching for the convex cone of the spectral data (see [10, 9] for an overview), and the other is through the optimisation of the sparsity of the abundance "a" [34] or using greedy algorithm for learning the dictionary D_u.

Classical search methods that exploit the convex distribution property of data are further sub-divided into algorithms that assume that pure pixels are in-scene such as the vertex component analysis (VCA) [35], and algorithms that make no such assumption of the pure pixels present in-scene like the more recent minimum volume simplex analysis (MVSA) [36] algorithm and the Collaborative Nonnegative Matrix Factorisation (CoNMF) [37] algorithm. These class of algorithms have provided good solutions especially when relatively pure pixels are present in the scene. The VCA, MVSA and the CoNMF algorithms have been employed here as competing HSI reconstruction methods to compare with the proposed algorithm. The decision for the selection of the type of algorithm for the given type of data is illustrated in figure 2.2.

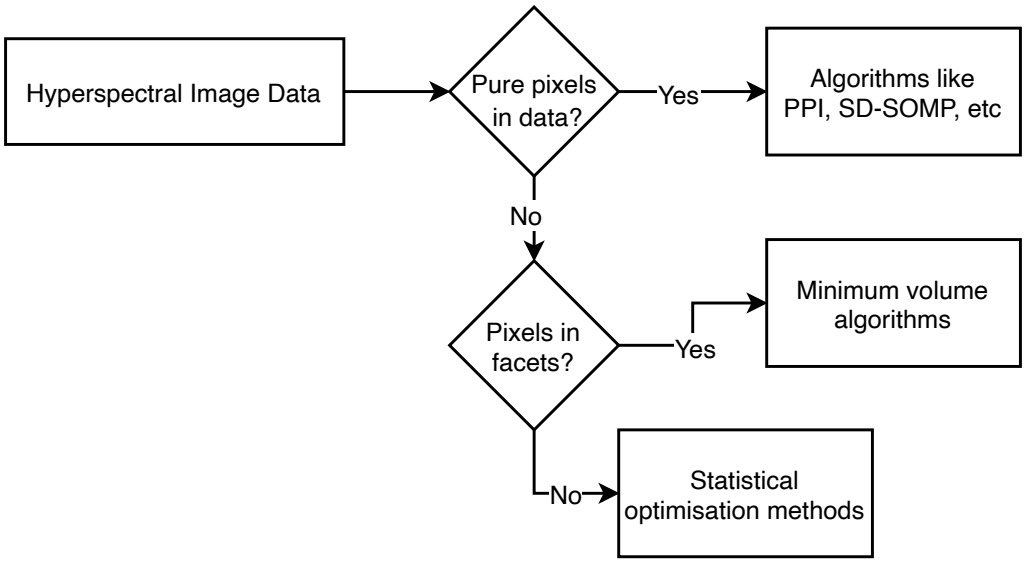


Figure 2.2: Different approaches to spectral unmixing

2.1.3.1 Unmixing assuming pure pixels are in-scene

The first class of unmixing algorithms are the ones that assume pure pixels are present in the simplex data cloud. These algorithms are relatively fast compared to other methods, but are weak performing due to the selection of only one sample per class and susceptible to the noise and errors in that representative sample. The Pixel Purity Index (PPI, [38]) algorithm and the Vertex component analysis (VCA, [35]) are well-known algorithms in this class, and, more recently Self-Dictionary with Simultaneous Orthogonal Matching Pursuit (SD-SOMP, [14]) algorithm.

The PPI algorithm projects a vector from a dictionary matrix of orthogonal vectors, called skewers, to the image data and selects the pixel most correlated with the given skewer. The skewers are initiated randomly with the number of skewers are taken as an user-input or determined by algorithms such as Virtual Dimensionality (VD, [39]) and hyperspectral subspace identification by minimum error (HySime, [40]), which may result in different candidate pure-pixel selection from one run to another. The mini-code of the PPI is written down in algorithm 1.

Algorithm 1 Pixel Purity Index (PPI) algorithm

1: **Import** HSI Image as 'Y', number of skewers 'n'
2: **Initialise** skewer dictionary as 'D' as 'n' normal random numbers
3: **for** `k = 1 to n` **do**
4: $x = D_k^T Y$ ▷ Project skewer D_k to all pixels
5: **Set** position of $\max(|x|)$ to p
6: EM $\leftarrow$ EM $\bigcup$ Y(p) ▷ Add Y(p) to EM dictionary

The N-FINDR [41] is another method in this class whose objective is to iteratively locate the purest pixels. The algorithm begins with a set of random selection of pixels and works its way towards the purest pixels located, which occupy the simplex with the maximum volume. Given a set of 'n' number of EMs = $\{e_1, e_2, e_3, ..., e_n\}$, the volume of a simplex is calculated by:

$$V = \frac{det(|\beta^T \beta|)^{0.5}}{(n-1)!}, \text{ where } \beta_i = e_{i+1} - e_i \tag{2.7}$$

Compared to PPI and N-FINDR algorithm, for a given 'i'th iteration, the Vertex Component Analysis (VCA) starts with the pixel with the maximum magnitude with ℓ_1 norm which by LMM T1 must be a pure pixel, and then projects a randomised vector (say v) in the space orthogonal to the EM dictionary selected in 'i-1' iterations and selects the pixel that is extreme in the projection i.e. the pixel 'y' with the maximum value in $v^T y$.

Self-Dictionary with Simultaneous Orthogonal Matching Pursuit (SD-SOMP, [14]) is a relatively recent algorithm in the class of unmixing algorithms with pure pixel assumption. Both VCA and SD-SOMP assumes that the EMs are in the vertices of the simplex. The algorithm like VCA starts with the pixel with the maximum magnitude, and moves on to select the next pixel with the maximum SOMP residue until the residual error for pixel in 'i'th iteration is within desired tolerance. The residual error for stopping criteria is calculated with the 'i'th pixel from EM dictionary selected in 'i-1' iterations, where the abundance is estimated by FCLS. SOMP residue for EM dictionary 'D_u' and sample 'y' is given by:

$$r = (I - D_u D_u^{-1})y \tag{2.8}$$

The residual error tolerance in SD-SOMP algorithm is estimated with HySime. The pseudo-code of SD-SOMP is written in algorithm 2.

Algorithm 2 SD-SOMP algorithm

1: **Import** HSI Image as 'Y' with 'b' number of bands, number of EM dictionary as 'n'
2: **Set** position of $\max(\sum_b |Y|)$ as first EM dictionary
3: **for** `k = 2 to n` **do**
4: r = (I - DD^{-1})Y ▷ SOMP residue
5: **Set** position of $\max(\sum_b |r|)$ to p
6: EM $\leftarrow$ EM $\bigcup$ Y(p) ▷ Add Y(p) to EM dictionary

The recent SD-SOMP will be used for comparison with the proposed methods in our experiments, along with the popular VCA unmixing.

2.1.3.2 Unmixing assuming pixels are in facets

The second class of SU methods are the ones that make no assumption that there are pure-pixels in the scene, but the assumption is that pixels are on the facets of a given simplex. Minimum volume algorithms belong to this class of SU methods, where the objective is to find the EMs that occupy the least volume in a simplex. The Minimum Volume Simplex Analysis (MVSA) like in [42] used VCA to subset the sample space Y. With the VCA EMs selected, search for true EMs were solved by minimising for the simplex volume encompassing true EMs. This is done by assigning a variable Q, which is the inverse of EM ($Q = EM^{-1}$) and solving the following problem:

$$Q^* = arg\max_{Q} \log|det(Q)| \text{ s.t. } QY \geq 0 \text{ and } 1^T QY = 1 \tag{2.9}$$

MVSA in [42] uses sequential quadratic programming (SQP) to solve the problem. It was amended in [36] by solving the problem with interior point method instead of the SQP to reduce computational complexity. The updated method for MVSA will be used for comparison in our experiments.

2.1.3.3 Unmixing as optimisation problem

The third class of unmixing are the ones that do not place any assumption to the location of mixed pixels in the simplex. This class of unmixing approximates EM by solving through a least squares optimisation problem, and the EM outcome may not be unique from one run to another. Non-negative matrix factorisation (NMF) methods and regression methods like [43, 44] are used in this space. Alternating least squares NMF is a common method in decomposition of a matrix 'X' to 'U' and 'V' is solved by alternating between the two steps below until convergence:

- The first step is to solve for U first by fixing the rows of the matrix U and solving for the columns of the matrix V, and
- The second step is to solve for U by fixing the columns of the matrix V and solving for the rows in the matrix U.

Collaborative methods are methods that solve more than one objectives at the same

time, and collaborative NMF (CoNMF) in [45] approximates both the EM dictionary and the abundance at the same time. The CoNMF achieves this with two regularisers: one to promote sparsity on the abundance matrix, and the second to promote minimum volume. The algorithm is updated in [37] modifying the second regulariser to be more robust to noise. The updated method will be used for comparison in our experiments. Much of the algorithms in this class are in common with dictionary learning algorithms which is discussed in the next section 2.2. A summary of the unmixing methods is presented in table 2.1.

Table 2.1: A summary of spectral unmixing algorithms

	Assumption		
Algorithm	**Full pixel in-scene**	**Simplex**	**Approach to solving LIP**
PPI	Yes	Yes	Projects skewer vectors to the scene
N-FINDR	Yes	Yes	Slowly iterates to the pixels that represents a simplex occupying the maximum volume
SD-SOMP	Yes	Yes	Pixel with the maximum SOMP residue is the next pure pixel
VCA	Yes	Yes	Selects pixel with the highest magnitude orthogonal to the previously selected EMs
MVSA	No	Yes	Minimum volume method that uses SQP/ interior point method to find EMs
CoNMF	No	No	Approximates EM dictionary and abundance at the same time

2.2 Dictionary learning (DL)

2.2.1 Principle

Dictionary learning is a branch of image and signal processing that aims to find linearly independent basis in a n-dimensional data cloud. Every pixel/data point in that cloud can be expressed in a sparse (mostly zeros) linear combination (called representation) of the basis. The database/collection of the basis is called a dictionary, and the sparser the representation, the better the dictionary. The basis set (similar to unit vectors in higher dimensional space) contains orthogonal atoms which can be obtained by various mathematical optimisation methods, and each symbolises an unique direction (vector) in the presence of measurement noise.

Mathematically, in a 'b' dimensional real space, a signal or a pixel 'y' (y $\subset$ Y collection of samples) can be recovered by measurement dense matrix called a dictionary 'D', and a representation 'a' which is assumed to be sparse (and ideally non-negative). The entities in a dictionary, called atoms, are ideally independent of each other and number of atoms in 'D' is far less than the total number of samples 'Y'. The objective function 'J' for dictionary learning (DL) is given by:

$$J(D,a) = ||Y - Da||_F^2 + \lambda ||a||_0 \tag{2.10}$$

Modelling a given signal or pixel as a sparse linear combination of atoms drawn from a dictionary (a database) is known as sparse coding. Sparse coding, which is also another type of matrix factorisation, has become popular in image and signal processing, e.g. [46, 47]. The most common approach to solving the DL is in two sub-steps [48]:

- The first sub-step is **sparse approximation**, where the representation 'a' is updated with the 'D' fixed from previous iteration
- The second sub-step is the **dictionary refinement** step, which is to update the 'D' keeping 'a' fixed. The two sub-steps are shown in figure 2.3.

Two types of methods solve the representation problem: greedy iterative algorithms that select one atom at a time based on the residual error in previous iterations like the popular Orthogonal Matching Pursuit (OMP) with some recent uses in [49, 50, 51, 52], and convex relaxation methods which solve the DL representation as a convex problem minimising for ℓ_p norm (p $>$ 0) [53, 54].

2.2.2 Sparse Approximation

2.2.2.1 Greedy Iterative Algorithms

Recent advances in dictionary learning (DL) have been made using greedy approaches such as orthogonal matching pursuit (OMP) [55], which finds the potential elements in a

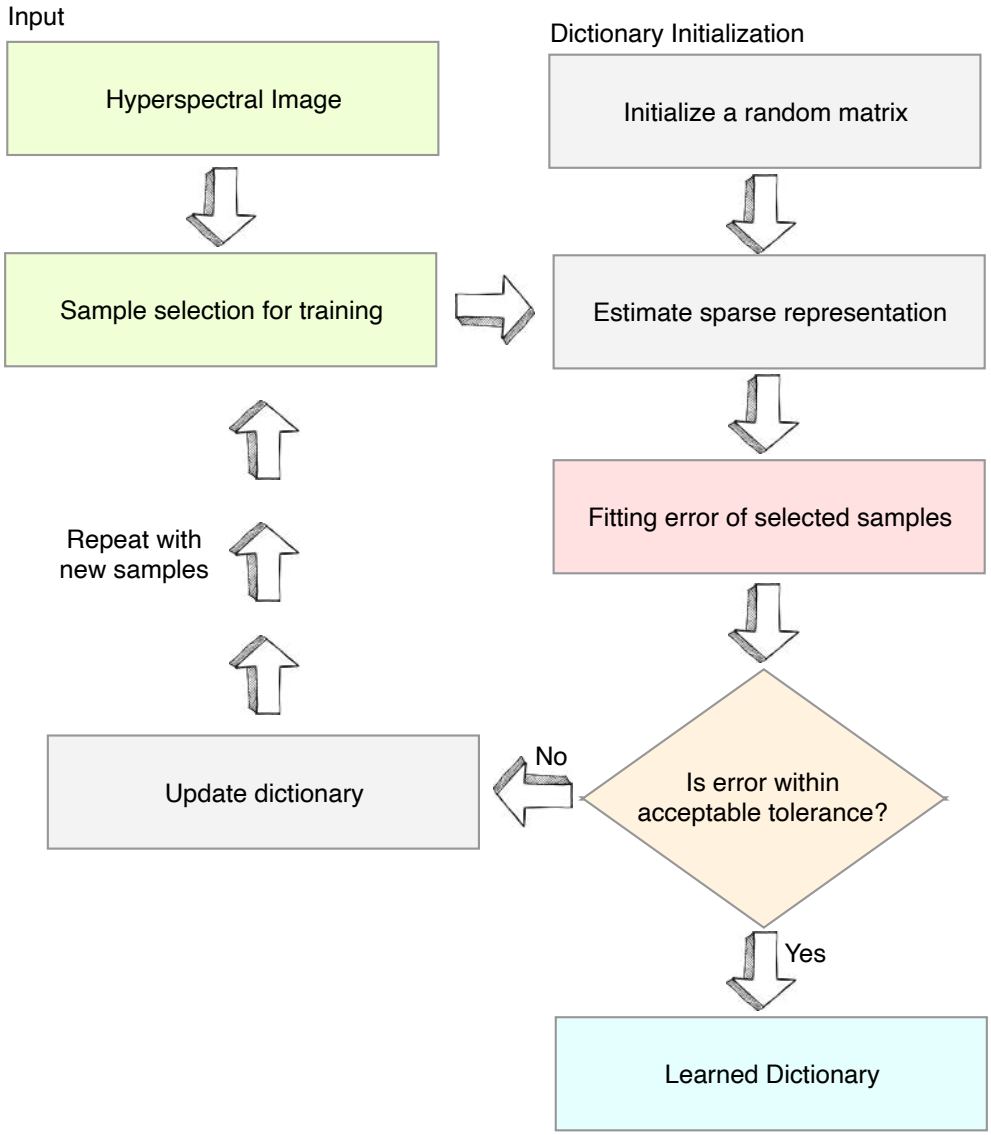


Figure 2.3: Flow diagram of common dictionary learning algorithms estimating the representation in its first step and then updating the learned dictionary for each training iteration. [3]

dictionary iteratively. New atoms are introduced whenever the residual has the largest absolute inner product between the dictionary atoms and the residual error, and the residual is subsequently modified, and the process is repeated untill the residual error is within acceptable tolerance. All potential atoms found are grouped into sub-dictionaries, S, and new atoms are selected orthogonally into subset 'S' in subsequent iterations.

Orthogonal Matching Pursuit (OMP)

Orthogonal Matching Pursuit (OMP) approximates sparse representation for a signal iteratively selecting one dictionary atom at a time that is most correlated with the signal's residue. OMP achieves this in 2 sub-steps (algorithm 3) that keeps repeating till the residue is within error tolerance:

- The first sub-step is to estimate the correlation of the atoms with the signal residue 'r', followed by selection of the atom with maximum correlation i.e. atom position with $\max(|D^T r|)$. The selected atom gets added to the selected subset of atoms from previous iterations.

- The second sub-step is to estimate the representation of with the selected atoms (usually by least squares) and update the residue by $r = y - Da$.

Algorithm 3 Orthogonal Matching Pursuit (OMP)

Input signal samples/pixel 'Y', dictionary 'D', maximum number of atoms per sample/pixel 'm', and 'n' noise/error tolerance.
Let 'S' be the total number of signal samples/pixels, and representation matrix 'a' be a zeros matrix with dimension 'N' number of atoms in D by S
for j = 1 to S **do** ▷ iterate for all samples
    $r_j = Y_j$
    Let $k = 1$ ▷ inner loop counter
    while $k \leq m$ AND $\sum |r_j| \leq n$ **do**
        $\phi_j = |D^T r_j|$
        $p_k \leftarrow \max(\phi_k)$ ▷ position of maximum
        $p_j \leftarrow \{p_j, p_k\}$
        $D_j = D(p_j)$ ▷ subset D to matched atoms
        $a(p_j, j) = (D_j^T D_j)^{-1} D_j^T y_j$ ▷ by least squares
        $r_j = y_j - Da_j$ ▷ update residue
        $k = k + 1$
Output representation 'a'

Several variants of OMP exist in efforts to speed up the algorithm, like, the generalized OMP (g-OMP) [56] selecting multiple atoms at the same time based on descending order of atomic correlation with the selected signal sample. Other variants like fast OMP (f-OMP) [57], Cholesky OMP and batch OMP in [58] using Cholesky or QR update process to reduce the computation of the dictionary inverse. The common factor between all these variants of OMP is that OMP is estimated on *per-signal case*, imposing a massive computational cost in terms of execution time for large datasets like AVIRIS hyperspectral images [59]. The computational cost is dependent on how the dictionary 'D' is inverted. For a D matrix with a size of 'M' by 'N', the inversion of D can be done by the following ways:

- Using Cholesky method by splitting up the D to a lower triangular matrix (say L) and its conjugate transpose (L^T). This method has a drawback in handling near-singular and rank deficient D and has a complexity of $O(MN^2)$.
- With QR decomposition where the D is split into an orthonormal matrix Q and an upper triangular matrix R. This approach is more stable than Cholesky and has a complexity of $O(N^3)$.
- By Moore–Penrose inverse decomposing the D by Singular Value Decomposition (SVD). This method is the most stable method among the three approaches and has a complexity of $O(N^3 + MN^2) \approx O(N^3)$ when $N \gg M$.

Non-negative OMP (NNOMP)

The objective of NNOMP group of algorithms is for all elements of the representation 'a' to be non-negative ($a_i \geq 0$, $a_i \in$ a). Similar to OMP, NNOMP is a two step process: atom selection and estimation of the coefficient of selected atoms. The differences in the process of NNOMP with OMP are:

- In the atom selection step, NNOMP selects the atom that has the highest positive correlation with the residue instead of the atom with the highest correction in case of OMP i.e. in a dictionary 'D' and residue 'r', NNOMP selects the atom with $\max(D^T r)$ while OMP selects the atom with the $\max(|D^T r|)$ ($|.|$ denoting the absolute value).

- The representation during each step of the iteration in NNOMP is estimated with non-negative least squares (NNLS) function [60], where NNLS is usually estimated by Lawson and Hanson method [61].

The recent update to the NNOMP algorithm is with fast non-negative orthogonal matching pursuit (FNNOMP, [57]), which combines both atom selection and non-negative representation estimation in one task. The FNNOMP algorithm does this with the QR factorisation instead of NNLS to obtain the representation, and has reduced the complexity of DL to O(NKlog(P)), where P is the inner loop count and P$\ll$K .

2.2.2.2 Convex Relaxation Algorithms

The DL problem of minimising the objective 'J' in equation (2.10) is a non-convex problem by itself, whose convexity comes from the sparsity promoting ℓ_0 pseudo-norm, and the bi-linearity between the dictionary 'D' and the representation 'a' [48]. The ℓ_0 norm is a pseudo-norm because the magnitude of the norm does not increase with increasing values of the elements in 'a'.

Basis pursuit methods solve the sparse decomposition problem by converting the ℓ_0 to ℓ_1 minimisation (L1LS). The L1LS problem is also known as the least absolute shrinkage and selection operator (LASSO). The LASSO solves the objective the same way by minimising the sum of squares error, but with an added ℓ_1 penalty compared to ridge regression with ℓ_2 penalty. An illustrative two-dimensional example geometry of different ℓ_p norms when ℓ_p = 1 is shown in figure 2.4.

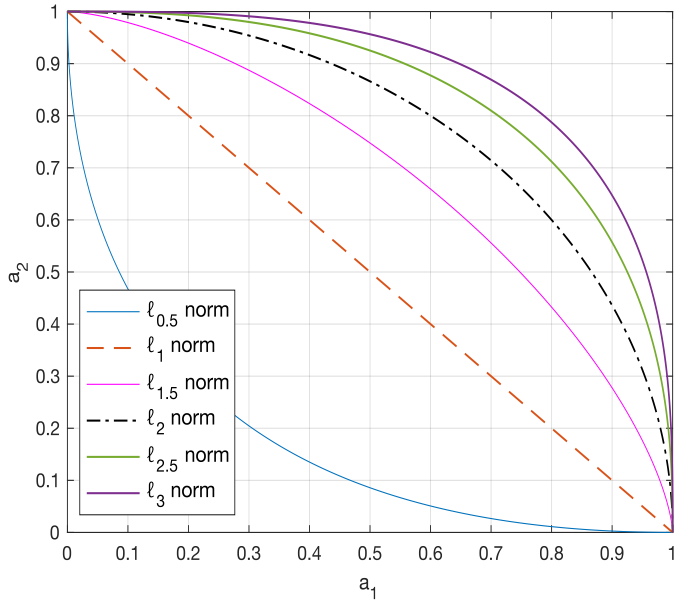


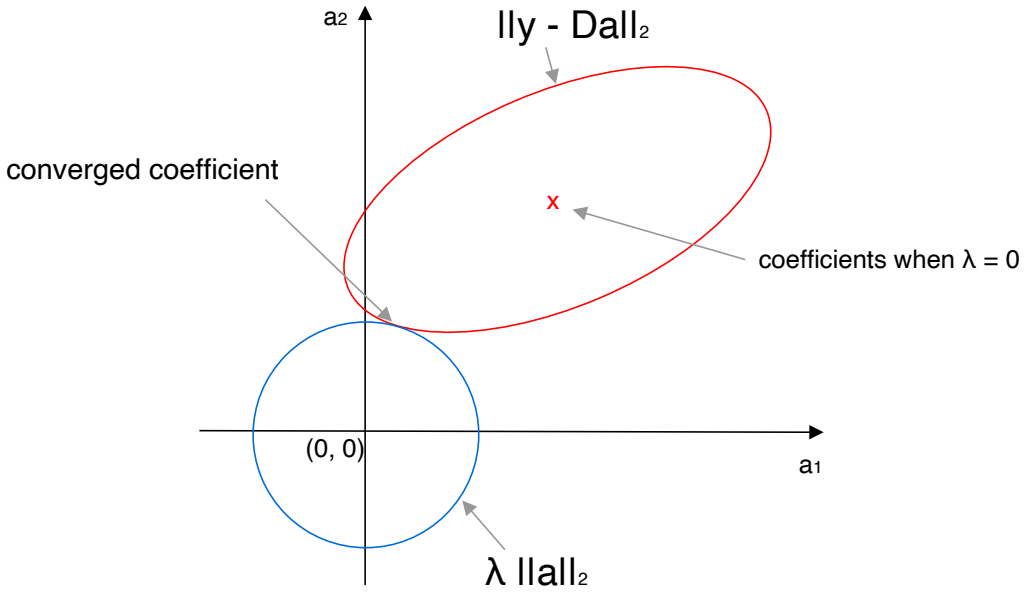
Figure 2.4: A two-dimensional illustrative example showing geometric shape of different norms in $||a||_p = 1$, when the value of 'p' varies from 0.5 to 3.

Minimising the sum of squared residual error by ordinary least squares which is the $||y - Da||_2^2$ term for a sample 'y' minimises the bias (the fitting error of the training set) but produces a high variance (the residual error of the test set). A low bias and high variance conclude an overfitting case, and introducing a penalty minimises overfitting. The differences between LASSO and ridge regression for a sample 'y' is in the penalty function with regularisation parameters λ_R and λ_L ($\lambda_R \neq \lambda_L$) for ridge and LASSO regression respectively, which are:

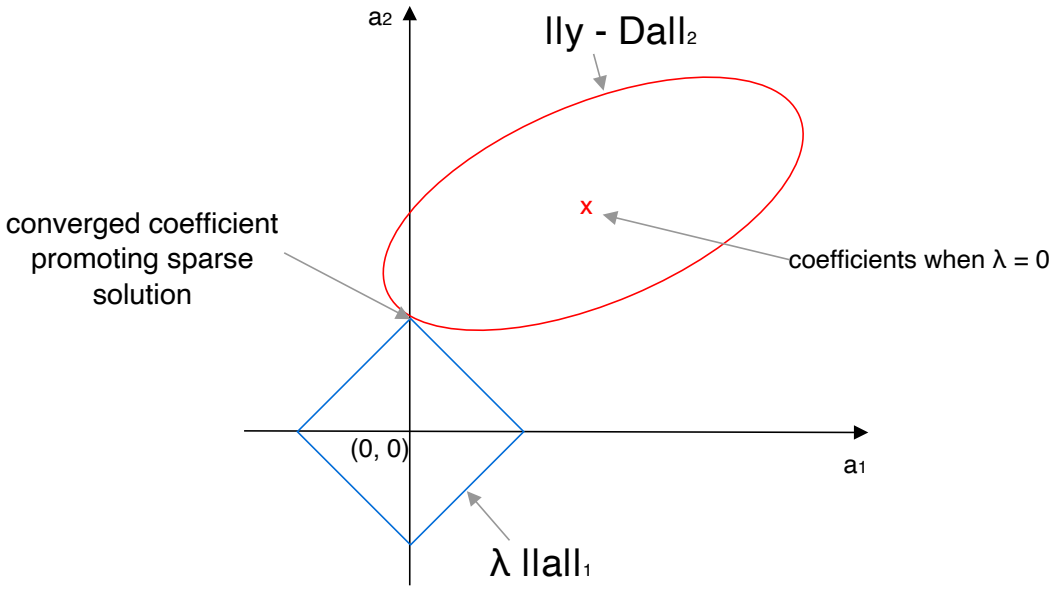
$$\text{LASSO: } ||y - Da||_2^2 + \lambda_L ||a||_1 \tag{2.11a}$$

$$\text{Ridge: } ||y - Da||_2^2 + \lambda_R ||a||_2 \tag{2.11b}$$

The penalty function of ridge regression scales the coefficients by a constant factor, whereas the LASSO translates the coefficients, truncating at zero thus promoting sparse solutions. Graphically, this can be visualised as shown in figure 2.5. And, elastic net regularisation linearly combines both LASSO and ridge, is more computationally expensive due to the two unknown regularisation terms.



(a) Ridge Regression



(b) Lasso Regression

Figure 2.5: A two-dimensional illustrative example showing the differences between ridge and lasso regression, where lasso is sparsity promoting.

Iterative thresholding algorithms is an approach to solve the Lasso problem. Hard thresholding algorithm is a two step process which first takes a gradient descend step on the representation 'a', then applies a hard threshold to the 'a' to select the 's' atoms with the largest magnitude. This two step iterative process continues till the desired error tolerance is met. A recent example is the paper in [62] applying a global hard threshold to the least squares function in order to select the 's' atoms with the largest magnitude in

representation among a total number of 'd' atoms in dictionary (s $<$ d) simultaneously for several signal samples/pixels. The mini-code of iterative hard thresholding is written in algorithm 4.

Algorithm 4 Iterative Hard Thresholding Algorithm

1: **Import** signal sample/image as 'y', dictionary 'D', thresholding operator 'H' to select 'k' sparse atoms, step size 's', and acceptable error tolerance 'e'
2: **Initialise** representation 'a' as 0
3: **While** $||y - Da||_2^2 \leq$ e
4: a = a + sD^T(y - Da) ▷ Take a gradient descend step on 'a'
5: a = H_k(a) ▷ Apply the threshold to select 'k' sparse atoms
6: **Output** representation 'a'

2.2.3 Dictionary Refinement

Once the representation is updated, the second sub-step is to update the dictionary 'D'. Least mean squares (LMS, [11]), K-Singular Value Decomposition (K-SVD, [13]), and Iterative least squares (ILS, [63]) are common approaches that update the 'D'. The LMS approach takes a gradient descend step following a mean squared error loss function of the linear model, which is represented in equation (2.12), where 's' is the learning rate or step-size with a linear decay in Adam's algorithm.

$$D = D + s\underbrace{(Y_s - Da)a^T}_{\Delta \text{ J(D, a)}} \tag{2.12}$$

The K-SVD is another popular DL algorithm whose main contribution is the dictionary update step. In K-SVD, the update step only uses the samples Y that makes a contribution to the representative atom. The atom update step is solved by SVD decomposition. The pseudo-code in algorithm 5 shows the K-SVD DL method.

Algorithm 5 K-SVD Dictionary Learning

Import HSI Image as 'Y', number of dictionary atoms 'n', and maximum number of iterations as 'M'
Initialize dictionary as 'D' as 'n' random numbers
for `k = 1 to M` **do** ▷ Iterate till convergence
    a = OMP(Y, D) ▷ Infer coefficients with OMP
    r = Y - Da ▷ residue
    for `j = 1 to n` **do** ▷ for all atoms
        $Y_j \subset Y$ and $r_j \subset r, \forall a_j > 0$ ▷ Subset samples that use the 'j'th atom
        T = $r_j + D_j a_j$
        [U, S, V] = svds(T, 1)
        D_j = U ▷ Update D_j with the largest contributing singular value
        $a_j = SV^T$ ▷ Update the representation
        $r_j = T - D_j a_j$ ▷ Update the residue
        $r \leftarrow r_j$

There is another family of algorithms that updates the dictionary at each iteration by least squares function, shown in equation (2.13). This class of algorithms is referred to as ILS-DLA [63], algorithm 6).

$$Y = Da \tag{2.13a}$$

$$\implies Ya^T = D(aa^T)$$

$$\implies D = Ya^T(aa^T)^{-1} \tag{2.13b}$$

Algorithm 6 Iterative Least Squares (ILS) Dictionary Learning

Import HSI Image as 'Y', number of dictionary atoms 'n', and maximum number of iterations as 'M'
Initialize dictionary as 'D' as 'n' random numbers
for `k = 1 to M` **do**
    $\min a$ in $||Y - Da||_F^2 = 0$ ▷ Infer coefficients with any method from section 2.2.2
    $D = (Ya^T)(aa^T)^{-1}$ ▷ Dictionary update by least squares
    normalise D

In summary, a dictionary learning approach is a combination of any row from table 2.2 matched with a row from table 2.3.

Table 2.2: Summary of various sparse representation methods

Algorithm family	Type	Description
OMP and its variants	Greedy	Selects the next atom from 'D' with the maximum '$D^T y$' value with pixel 'y'
Thresholding methods	Convex	Converts representation values to zero when it's below the threshold
LASSO	Convex	Sparsity promoting ℓ_1 regularisation

Table 2.3: Summary of the available methods to update a dictionary during an iteration

Method	Description
ILS	Updates all D at the same time by least squares
K-SVD	Updates D one at a time with SVD decomposition
LMS	Gradient descend step following a mean squared error loss function

C-SCD algorithm uses the LASSO and LMS update method which is covered in the next section 2.3.

2.3 Classic Sparse Coding Dictionary (C-SCD)

The Sparse Coding Dictionary (SCD) is a well known approach that decomposes the HSI into a linear array of a few bases and a sparse matrix. The bases or atoms are constrained to unit ℓ_p norm (generally ℓ_2), and the representation is a sparse matrix. The collection of atoms in a given scene is collectively referred to as the learned dictionary. SCD has an advantage when compared with the search method in that it is capable of finding the dictionary even when pure pixels do not exist in the scene, with a wide range of applications from classification to super-resolution [12, 64, 65]).

2.3.1 Principle of the C-SCD algorithm

One such SCD algorithm is proposed by Adam Charles et al. in [11] referred to as classic sparse coding dictionary (C-SCD) in this thesis builds on the work done by Olshausen and Field in [66, 67] and ports their algorithm to HSI. Olshausen and Field proposed to estimate prior probability distribution over the sparse coefficients 'a' given a set of basis ϕ (dictionary D in this thesis is ϕ in Olshausen and Field's paper), the

probability of an image 'I' (signal samples/pixel 'Y' in this thesis is image 'I' in Olshausen and Field's paper) arising from the model, graphically represented in figure 2.6 is mathematically written as:

$$P(I|\phi) = \int P(I|a,\phi)P(a)\delta a \tag{2.14}$$

where $P(I|a,\phi)$ is the probability of I arising from coefficients 'a', and P(a) is the prior probability distribution of the coefficients. The equation is graphically represented in an example in figure 2.6.

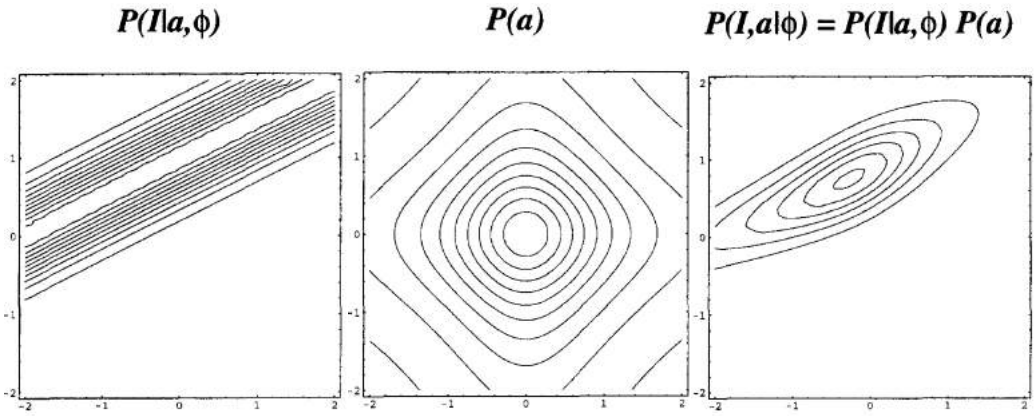


Figure 2.6: Example of two-dimensional iso-probability plot in Olshausen and Field's paper, where (a) is Gaussian likelihood, (b) is Cauchy prior, and (c) is the product of (a) and (b), refer [67]

The C-SCD algorithm brings Olshausen and Field's approach to hyperspectral imaging, solving the Lasso problem as a Bayesian approach. Bayes' rule for signal sample/pixel 'y' and representation 'a' is given by:

$$\text{Posterior probability } P(a|y) = \frac{\overbrace{P(y|a)}^{\text{likelihood}}\overbrace{P(a)}^{\text{prior}}}{\underbrace{P(y)}_{\text{normalising factor}}} \tag{2.15}$$

Using the Bayes' rule from equation (2.15), the un-normalised posterior is $P(y|a)P(a)$. The C-SCD algorithm assumes a Laplacian prior probability distribution on the representation 'a' due to the peak distribution around zero for sparsity, and a

Gaussian distribution on the likelihood, the un-normalised posterior for dictionary 'D', noise variance σ_e^2, and standard deviation on the prior σ_a becomes:

$$P(a|y) \propto \underbrace{e^{(-\frac{1}{2\sigma_e^2}||y-Da||_2^2)}}_{\text{likelihood}} \underbrace{e^{(-\frac{\sqrt{2}}{\sigma_a}||a||_1)}}_{\text{prior}} \tag{2.16}$$

The C-SCD then takes a negative logarithm on the posterior probability, and the maximum a posterior (MAP) maximising the posterior distribution results in the lasso regularisation (which contains ℓ_1 penalty) term 'λ_L' from equation (2.11) to be $2\sqrt{2}\frac{\sigma_e^2}{\sigma_a}$. Using this approach, the C-SCD algorithm has shown that learned atoms are very similar in shape to actual spectra and are capable of inferring HSI data from MSI images. The C-SCD model has been shown capable of reconstructing HSI scenes by using the dictionary that had been trained from imagery obtained from the same scene from another season.

2.3.2 Execution of the C-SCD algorithm

To recall, the objective function mentioned previously in equation (1.2), where 'D' is the atomic dictionary, representation 'a', sample 'y', and 'λ_L' the lasso regularisation term is written as:

$$J = \min ||y - Da||_2^2 + \lambda_L ||a||_1 \tag{2.17}$$

It is seen from the objective function that learning the SCD model is required to minimise both 'D' and 'a' at the same time. However, this cost function may not be jointly convex in the dictionary and the representation domains, which makes a global minimisation solution difficult. One solution to this problem is the implementation of a variational approach in the C-SCD algorithm, which alternates a minimisation with respect to the 'a' for the current 'D', then a gradient descent over the elements in 'D' is taken for the given calculated 'a' [11].

Following a random sampling of the materials to subset the training data (say $Y_k, Y_k \subset Y$ for training iteration 'k'), the first step of the algorithm is to estimate the sparse representation. One the representation 'a' is estimated, the second step of the

minimisation problem is to update the learned dictionary D for a given training iteration. C-SCD follows the least mean squares (LMS) algorithm to update the D.The C-SCD algorithm use least mean squares to update the 'D'. The LMS algorithm updates the dictionary atom in a stochastic manner with a mini-batch of samples, where the first derivative of the cost function 'J' w.r.t dictionary D is calculated by chain rule. The final step is that the updated dictionary is then normalised for all atoms of D to have a unit ℓ_2 norm. The complete mini-code of the C-SCD is presented in algorithm 7.

Algorithm 7 Classic Sparse Coding Dictionary (C-SCD) Learning Algorithm, refer [11]

1: **Import** HSI Image as 'Y', number of dictionary atoms 'n', user-defined step size 's' and step-size decay 'd', and maximum number of iterations as 'M'
2: **Initialize** dictionary as 'D' as 'n' random numbers
3: **for** `k = 1 to M` **do** ▷ Iterate till convergence
4: choose pixels 'Y_k' randomly ($Y_k \subset$ Y) ▷ Sample selection
5: min a_k in $||Y_k - Da_k||_F^2 = 0$ ▷ Infer coefficients
6: $D_k = (Y_k - Da_k)a_k$
7: $D \leftarrow D + sD_k$ ▷ Dictionary Update
8: s = s d ▷ Step size update

User-defined estimates like step size and decay, and least squares function to infer coefficients remain unchanged from [11]. An inner loop is required if more than one sample 'Y_k' is selected per iteration to estimate representation 'a'.

2.3.3 Unmixing with the C-SCD model

To learn EMs with the C-SCD model, two fundamental changes were necessary like in [34]. The changes are:

- the unit ℓ_2 norm constraint from atoms is removed because real EMs are not bound by such constraint, and
- the representation 'a' is constrained for sum-to-one abundance as per the linear mixing model (LMM).

To recall, there are two typical approaches to enforce the fully constrained least squares (FCLS) or T1 abundance condition required by CameoSim's material allocation module. FCLS for non-negative and sum-to-one abundance is enforced through two

Lagrange multipliers, one for positive constraint with Karush–Kuhn–Tucker condition and the another for sum-to-one. Alternatively, by using one Lagrange multiplier to maintain the positive condition and the sum-to-one condition is concatenated with ones vector in the input signals and the EMs. The latter method is more favourable in computation time as it does not need the second Lagrange multiplier. Further, a complete EM dictionary is used for LSMA and an overcomplete dictionary is used for MESMA.

With C-SCD unmixing solving the objective problem for the minimum mean error depending on the sparse representation method used to estimate the abundance, the proposed methods improves upon the C-SCD unmixing model and adopts it for an external material allocation for CameoSim simulator.

2.4 Chapter summary

Spectral unmixing (SU) is the approximation of true endmembers (EMs) in presence of near-full pixel materials (pure pixels) and subpixel materials (fractions of several materials within a single pixel). The competing methods of SU are a mixture of the classical unmixing method assuming a simplex and statistical form of unmixing methods. The three types of unmixing algorithms that have been studied here are:

- Simplex with in-scene pure pixel assumption such as the VCA algorithm due to its popularity in the remote sensing mainly because of its ability for estimating the Ems quickly. The SD-SOMP algorithm is a recently published matching pursuit algorithm which also rely upon the presence of pure EM pixel in the scene.
- Simplex without in-scene pure pixel assumption such as MVSA algorithm.
- Statistical method to select the EMs by optimising (minimising) the least residual error for the given training scene. Methods such as the C-SCD is a competing algorithm which uses Bayesian LASSO, and the CoNMF is a recent factorisation algorithm belong to this class of unmixing algorithms.

Chapter 3

CameoSim Simulator, Experimental Setup, and Assessment Metrics

This chapter provides the fundamentals of scene simulation process, specific to CameoSim simulator. This foundation is required to understand the workings of a simulator and the crucial part of material allocation. This chapter starts with section 3.1 by introducing CameoSim and its modules, followed by a thorough understanding of a simulator project's components in section 3.2. Section 3.3 details on file structure of material allocation. This chapter concludes by introducing the robust set of hyperspectral images used in this thesis for experimental work in 3.4, followed by the assessment metrics used to measure the accuracy of reconstructed data in section 3.5.

3.1 Introduction to CameoSim simulator

CameoSim (CAMouflage Electro-Optic SIMulation) is an independent simulator developed by Lockheed Martin UK for rendering general-purpose RGB colour, multispectral, and hyperspectral images. Hyperspectral scene simulation is one of the modules of the simulator, and this module serves the applications of several defence agencies across the globe, as seen from publications from the Swedish FOI in [68], the United States Air Force, and the Australian DSTO in [5].

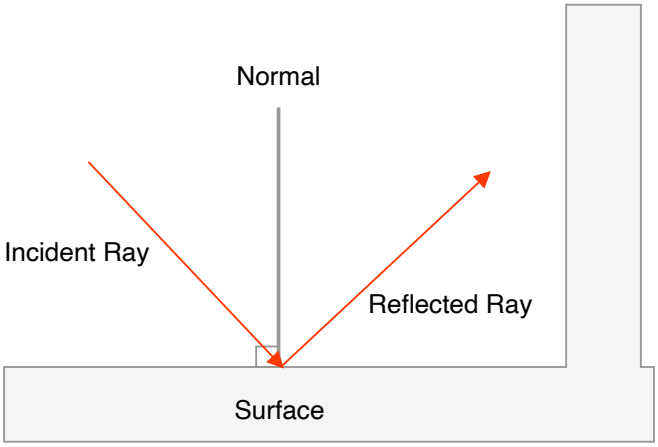
Scene simulators use principles of radiative transfer (RT) [69], and CameoSim

simulator renders RT by radiosity model, which consists of radiance along the target pixel-sensor path and radiance due to scattering of light. In a Monte-Carlo rendering scheme, the value of a pixel's measurement 'I' is measured in terms of an integral over the space 'P' of all light paths (an interpretation of how a ray of light would behave in real space) is given by:

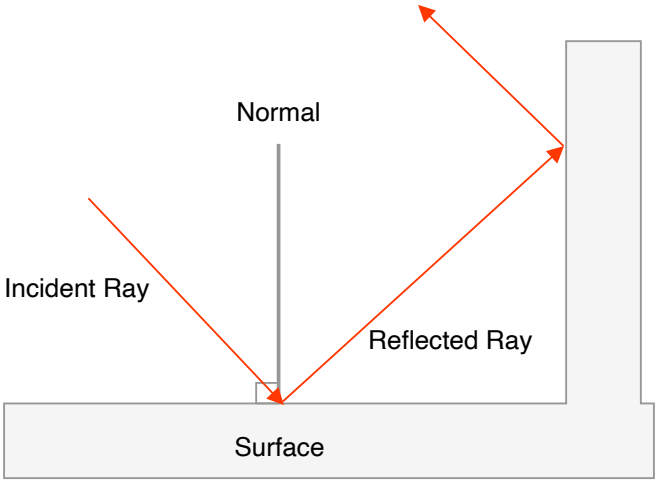
$$I = \int_P f(r)\delta r \tag{3.1}$$

where f(r) is the measurement contribution of light path. Due to the proprietary nature of the software and export control issues, the exact methodology implemented in CameoSim is unknown, however, the bouncing of light rays is controlled through radiosity settings which consist of:

- **Quick radiosity** is where a single incident ray bounces off the surface with a single reflected ray before it is observed by the sensor, and
- **Full radiosity** is where a single incident ray is allowed to bounce off of multiple surfaces. The differences between quick and full radiosity are shown in figure 3.1.
- And, an optional **true radiosity** toggle that applies to both quick and full radiosity. When true radiosity is turned off by default, a single incident ray results in a single reflected ray. The true radiosity toggle, if toggled on, allows a single incident light ray on hitting a surface to output multiple reflected rays following the Lambertian surface property, where the number of output rays is user-defined.



(a) Quick Radiosity



(b) Full Radiosity

Figure 3.1: Illustration showing radiosity options in CameoSim

CameoSim software package is one of the oldest commercial simulators that offers several modules, each with a collection of tasks serving an objective. Each module has a separate interface. The multiple functionalities and tools listed in figure 3.2 shows the available modules in CameoSim version 6.5, which is licensed to Cranfield University and is used for this research work. Among several modules in the software, this chapter describes only the essential components required to define scene simulation process, and how/where material allocation plays a role.

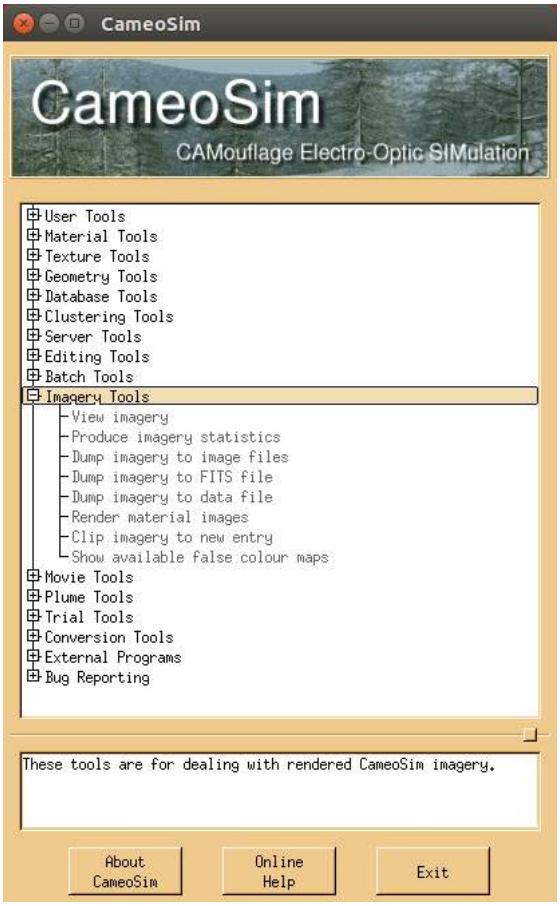


Figure 3.2: Different modules in CameoSim simulator

For hyperspectral scene reconstruction and simulation, the modules of interest from the figure are:

- **Texture tools**: contains a group of functions/sub-functions which are used to input/generate the material allocation LUT file. Material allocation is contained in this module.
- **Geometry tools**: is used to generate the 3D geometry of the simulated scene. Where 2D geometry is required, a flat plane in 3D space is placed to produce that effect.
- **Database tools**: contains a collection of all past and current simulation projects, with a collection of simulating parameters within each project.
- **Imagery tools**: is used to export a simulated image to common file formats.

3.2 Scene simulation project components

In the heart of CameoSim simulator is the database (DB) manager. The DB manager contains a list of all projects and generated atmospheres required for simulation. Once material allocation is performed in the texture tools module, the simulating parameters are brought together as a part of a single project file, stored in the DB window. The DB tools window, an example shown in figure 3.3, shows how the DB window presents itself with different projects stacked in a tabular manner.

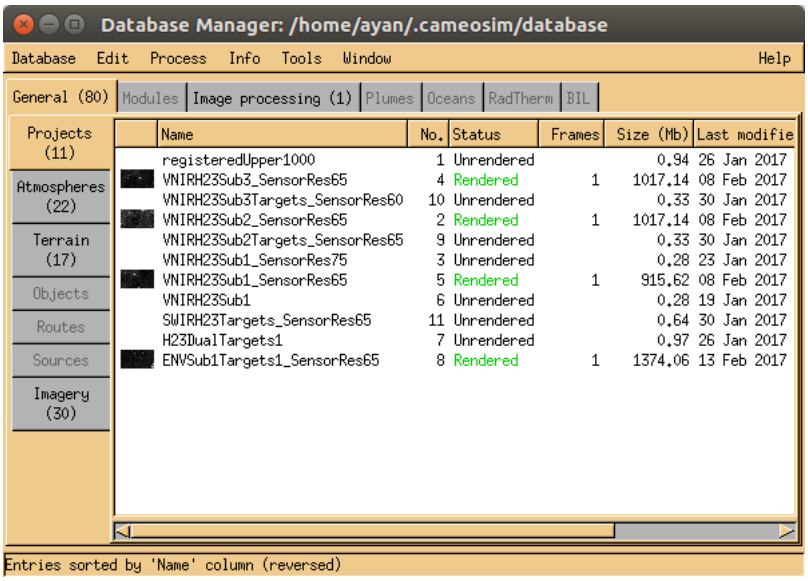


Figure 3.3: Database manager window with different simulation projects

A DB manager contains a list of all projects, and, a set of rendering parameters for scene simulation is collected under a single project. Each project has its own set of rendering parameters that aim to replicate a real-world HSI data acquisition process: (i) parameters that define the atmosphere, (ii) parameters that define the how high a given sensor is from the ground taking inputs such as sensor elevation and azimuth, and (iii) parameters that take input the spatial and spectral resolution information of the sensor. An example project window is shown in figure 3.4. Additional components that play a role in ray tracing are also taken as input within project window, like, sampling technique to be used, and use of computer resources where one defines the CPU usage limits.

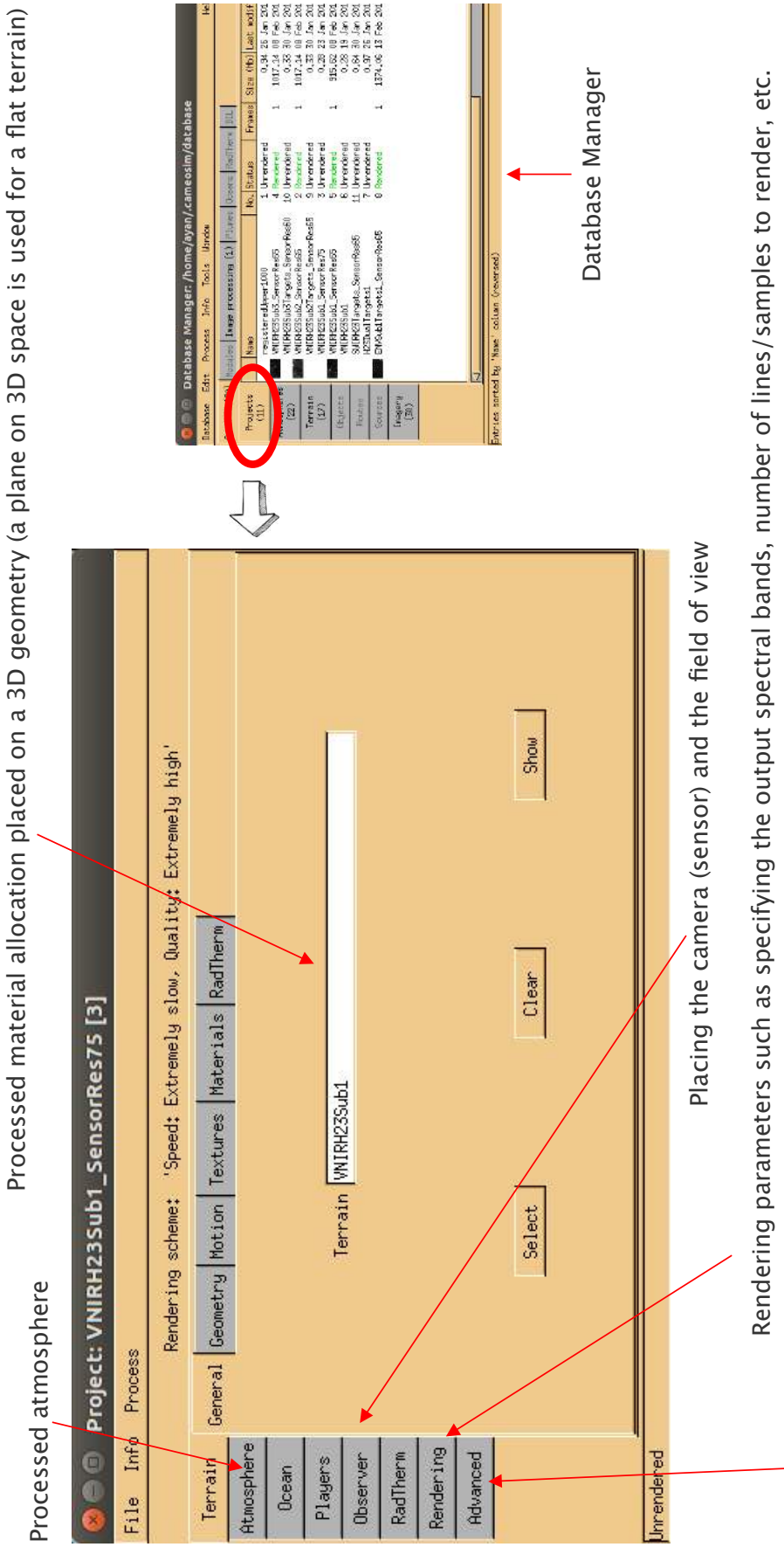


Figure 3.4: Project window for a single scene simulation project

3.2.1 Sampling

While the number of physical light rays shining on a surface may be infinite, CameoSim performs raytracing with Monte-Carlo sampling scheme and the principles of radiative transfer rendering equation to obtain the approximate radiance of a pixel. The three main components of the rendering equation are the thermal self-emission, the atmospheric terms, global illumination accounting for reflected radiance [69]. A physical ray of light is represented by the intensity value obtained by evaluating the rendering equation.

Sampling module in CameoSim suggests how many rays are fired and where they are fired to a single pixel. Depending on the sampling scheme used, a pixel is subdivided into subpixels and rays are fired according to the subdivisions. CameoSim implements one ray per pixel, regular grid, super-sampling, and Poisson sampling techniques.

3.2.1.1 One Ray Per Pixel sampling

In one ray per pixel scheme, a single ray fired onto a pixel. Given a pixel of 'h_p' height and 'w_p' width, a single ray is fired at the location

$$(\frac{h_p}{2}, \frac{w_p}{2}) \tag{3.2}$$

An optional parameter, jitter, is to introduce a random variation on the location of the light ray. For example, a jitter of 10% means that the light ray can deviate to a maximum 10%, or, a maximum of $\pm 5\%$ on each side. An input of 0% for jitter turns off the random variation. An example of one ray per pixel sampling is shown in figure 3.5.

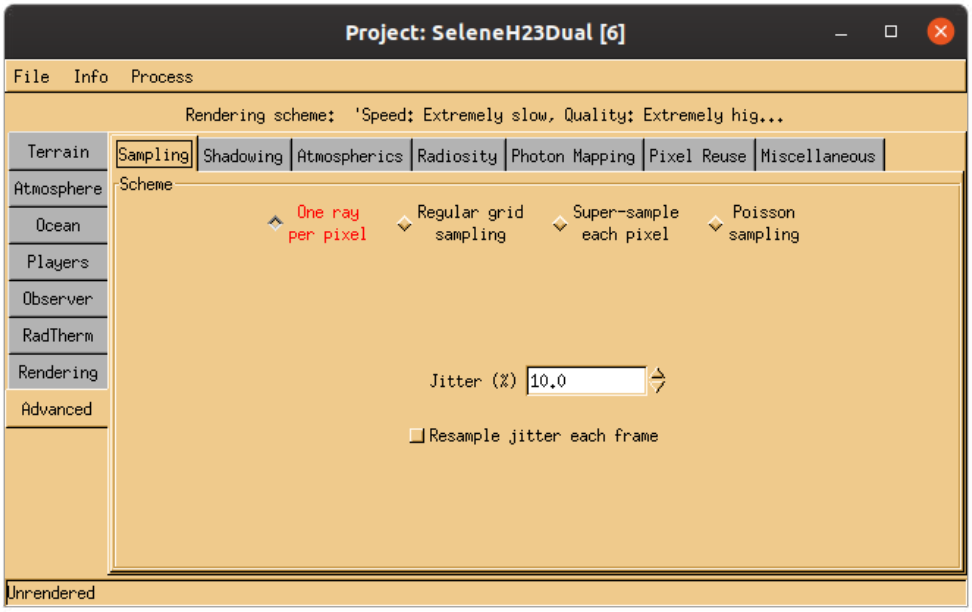


Figure 3.5: An example of one ray per pixel sampling scheme in CameoSim

3.2.1.2 Regular Grid sampling

The regular grid sampling technique is where a pixel, given 'n' number of divisions (where n is an integer), a pixel is divided by 'n' rows and 'n' columns, for a total of 'n^2' subdivisions. And then a single ray is fired onto a subdivision the same as one ray per pixel sampling, with the optional jitter. Figure 3.6 shows an example.

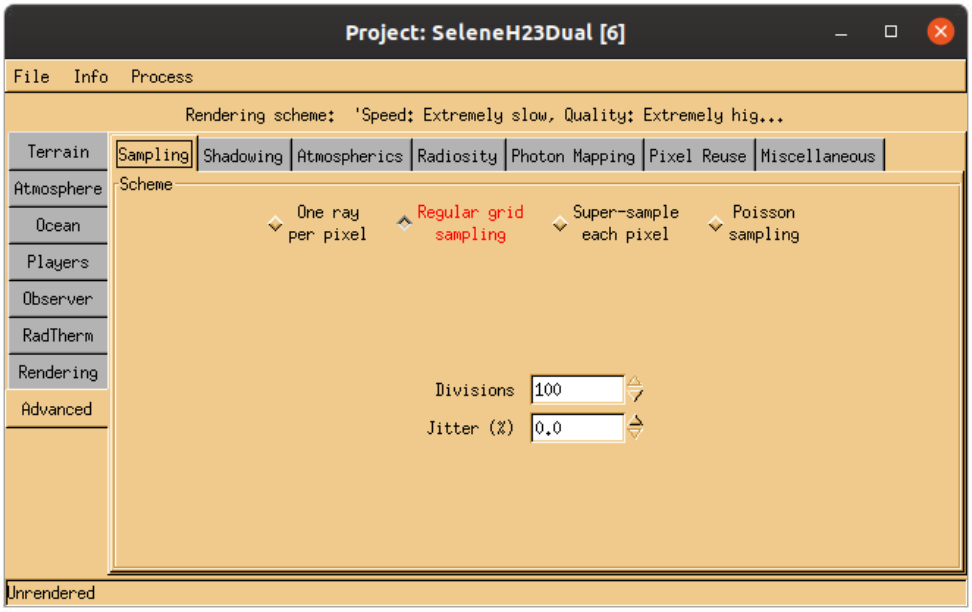


Figure 3.6: An example of regular grid sampling scheme in CameoSim

3.2.1.3 Super-sampling

Super-sampling scheme fires the number of rays depending on how much intensity differences within a pixel. It starts by sending a fixed number of rays to a pixel. CameoSim sends four rays with the optional jitter, subdividing a pixel into four grids at the start. If the resulting intensities of all rays are similar, i.e. looking at a single object, no further rays are sent to that pixel. If the colour intensities are of any two adjacent rays are greater than the user-defined threshold (in the input field labelled 'contrast'), a further subdivision is necessary, and the process repeats till the intensity differences between two adjacent rays are less than the defined threshold. Figure 3.7 shows an example.

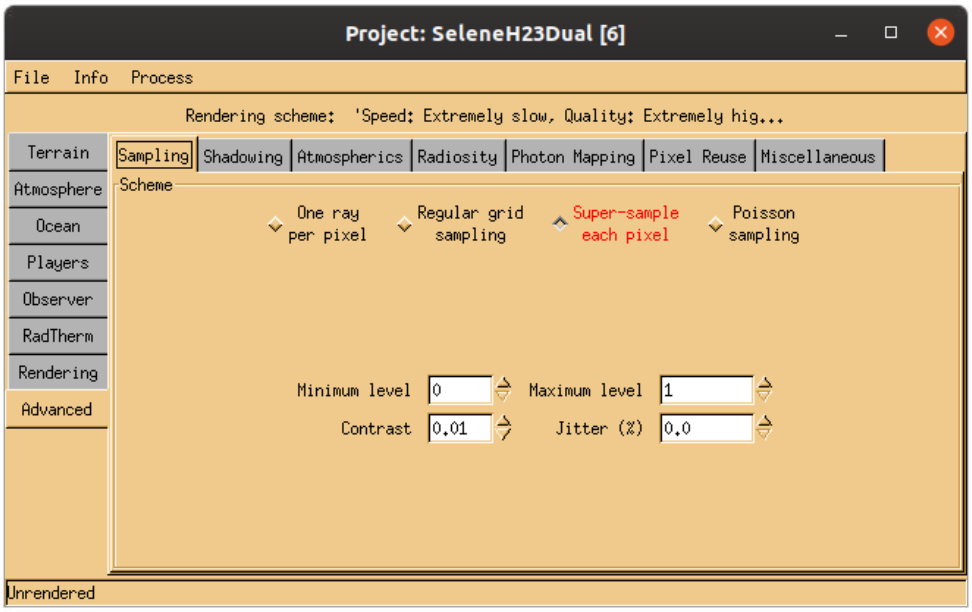


Figure 3.7: An example of super-sampling scheme in CameoSim

3.2.1.4 Poisson Sampling

Poisson sampling fires rays randomly according to Poisson distribution. The Poisson distribution with the mean of the distribution (λ) is given as:

$$P(x,\lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \qquad x = 0,1,2,\ldots \tag{3.3}$$

CameoSim first fires the user-defined number of sample rays (say 'k'), and checks for the intensity differences between adjacent rays. If the intensity differences (contrast input) is more than the user-defined acceptable threshold, another 'k' rays are introduced to the scene, making a total of 2k rays. The process repeats 'p' times for a total of 'pk' rays until the intensity differences are within defined threshold, or, if 'pk' is greater than the user-defined maximum rays per pixel. Figure 3.8 shows an example.

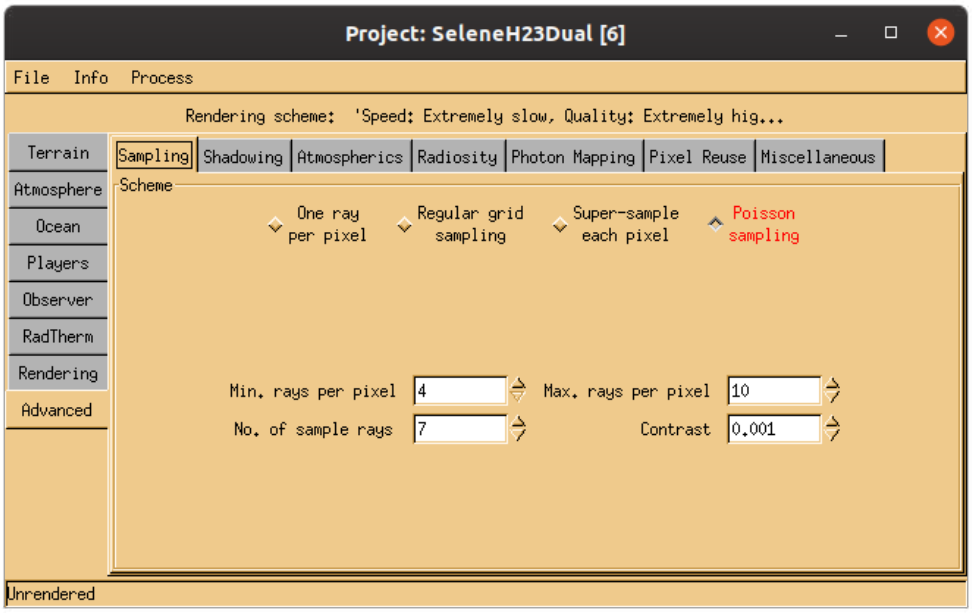


Figure 3.8: An example of Poisson sampling scheme in CameoSim

3.2.2 Defining the atmosphere

A scene requires a defined atmosphere for realistic rendering. CameoSim uses the well-established MODerate resolution atmospheric TRANsmission (MODTRAN) [70]. MODTRAN is a software designed to model atmospheric propagation. CameoSim communicates with MODTRAN to develop the required atmospheric components necessary for scene simulation. The atmosphere window is shown in figure 3.9 takes input for MODTRAN such as date, time of day, geographic location of the scene, and cloud-type parameters.

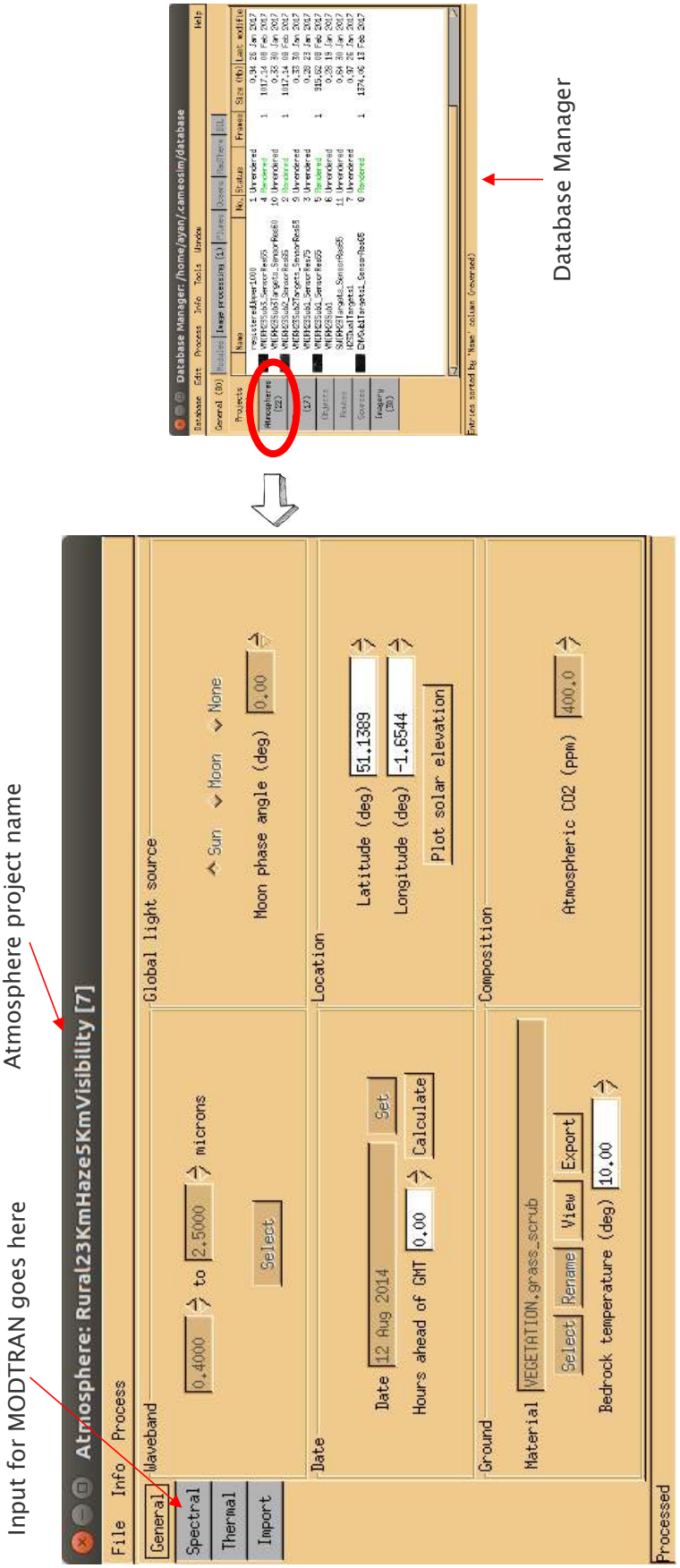


Figure 3.9: An example atmosphere window in CameoSim taking inputs for MODTRAN

3.3 File structure for material allocation

The accuracy of material allocation, or image decomposition to EM-abundance pair and its subsequent reconstruction, determines the accuracy of the reconstructed input reflectance to the simulator. This input reflectance is ray traced with radiative transfer equations to produce the output radiance. The quality of scene simulation increases or diminishes with improvement or decline in material allocation.

While CameoSim has in-built methods for material allocation, currently through RGB input, the proposed method (i) expands this module with external classification input from multispectral and hyperspectral images, and (ii) proposes a novel method for material allocation to be more robust to trace materials for detection applications, and with minimal error. CameoSim inputs the EM-abundance through the texture tools, shown in figure 3.10.

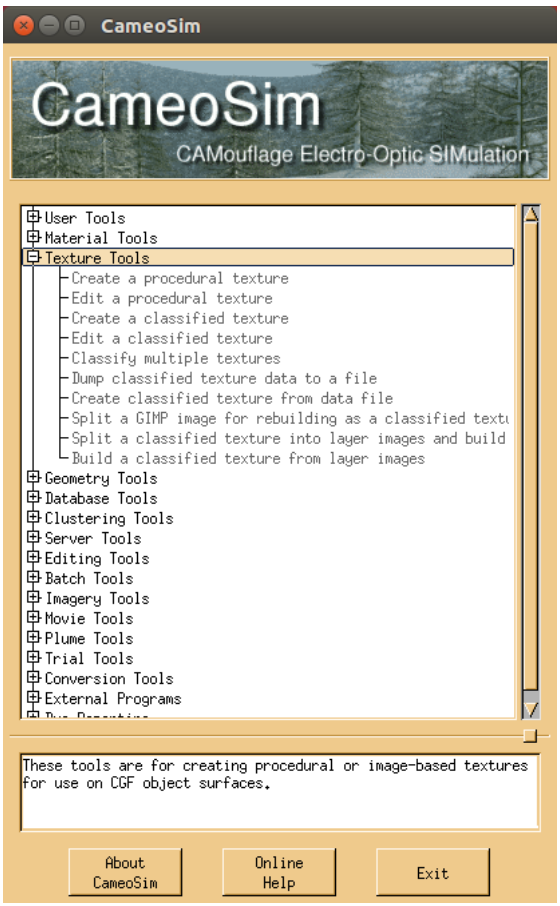


Figure 3.10: Texture tools in CameoSim for material allocation

CameoSim inputs EM-abundance through two files: (i) individual files for each endmember material with its reflectance and other optical properties, (ii) a lookup table (LUT) associating each EM material with its respective abundance values. The LUT input linearly mixes the endmember materials with their respective abundance values, with a maximum limit of four materials per pixel in the current structure. As with spectral unmixing, there is a sum-to-one abundance criterion for all pixels, irrespective of illumination differences. Figure 3.11 demonstrates the structure of the input file for material allocation.

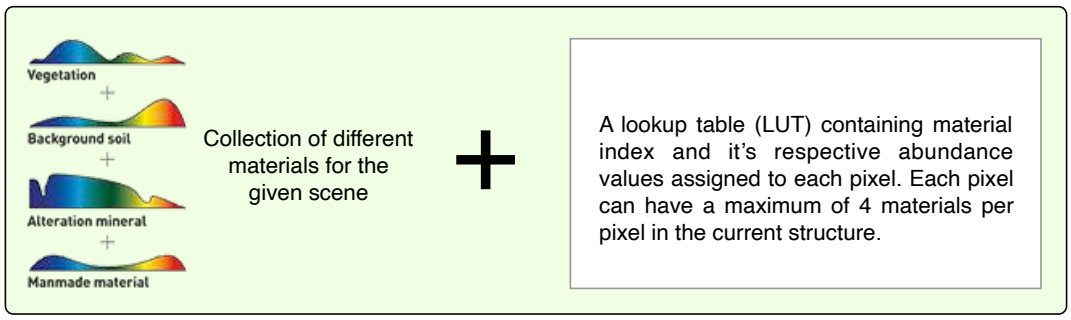


Figure 3.11: Structure of EM-abundance input used by CameoSim [3].

3.4 Hyperspectral images used in this thesis

Scenes from Selene dataset [71] such as 'Selene H23 VNIR' and 'Selene H23 Dual' are primarily used in this thesis. Selene is discussed in detail at a later part of this section in 3.4.1. Apart from the Selene, publicly available 'Paso Robles-Monterey' (AVIRIS dataset flight name: f150615t01p00r11, RGB image shown in figure 3.12) which is a high altitude AVIRIS imagery consisting of vegetation, highway, and cities.

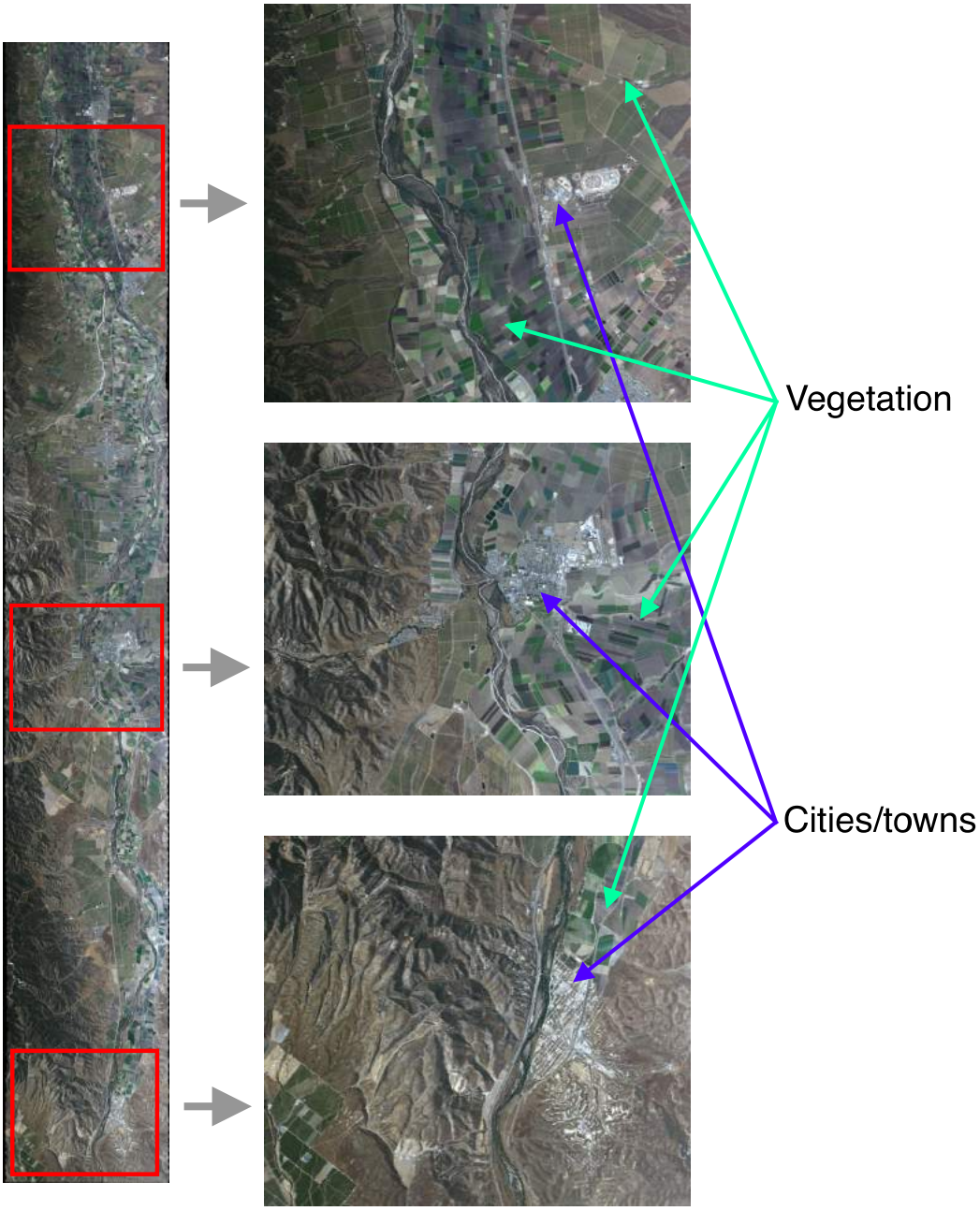


Figure 3.12: RGB image of Paso Robles-Monterey scene.

Additionally, three 'Virginia City' images (which were accessed from `https://www.spectir.com`, RGB image shown in figure 3.13), are images of a mountainous region. These three images have also been employed for experimental validation of the methods.

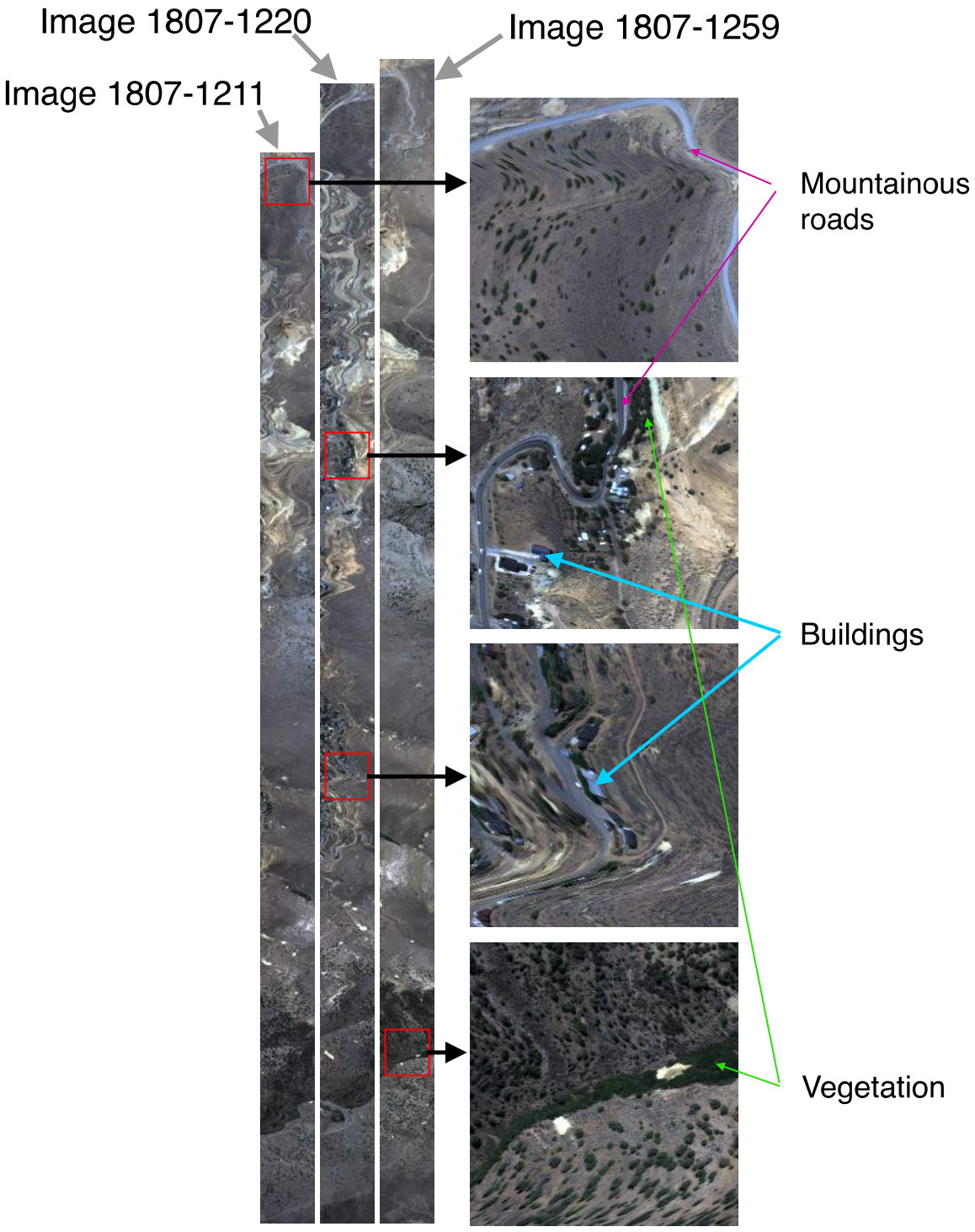


Figure 3.13: RGB image of Virginia City scene.

These datasets will be used to access the quality of existing and proposed scene reconstruction. Table 3.1 provides a summary of spatial and spectral dimensions of the different scenes.

Table 3.1: A summary of the hyperspectral scenes used in this thesis

Hyperspectral Images	**Lines**	**Samples**	**Bands**	**Spectral Range** (in μm)
Selene H23 VNIR	3752	1600	160	0.41 to 1
Selene H23 Dual	1876	380	448	0.41 to 2.5
Paso Robles-Monterey	5115	741	224	0.36 to 2.5
Virginia City 1807-1211	6349	320	178	0.4 to 2.45
Virginia City 1807-1220	6758	320	178	0.4 to 2.45
Virginia City 1807-1259	6904	320	178	0.4 to 2.45

3.4.1 Overview of Selene Scene

Selene dataset is the primary dataset of this thesis. This preference is primarily due to the availability of known subpixel target materials in the scene. The 'H23 VNIR' scene was acquired by HySpex VNIR-1600 sensor at Porton Down range (Long $51^{\circ}8'19.7''$N Lat $1^{\circ}39'16.9''$W to $51^{\circ}7'41.7''$N $1^{\circ}40'8.5''$W) on 12 August 2014 BST 12:00:04. Natural materials like grass, soil and tree cover over 95% of Selene scene, and artificial materials such as ground markers, path, concrete, building and coloured panels cover the remaining scene. Without any prior knowledge of the atmosphere, the Quick Atmospheric Correction algorithm (QUAC, [72]) was applied to the raw data using ENVI software with a generic sensor to obtain the reflectance of this dataset. The reflectance data obtained after QUAC algorithm is the 'ground truth' data used for reference when compared with material allocation/reconstruction methods.

The VNIR scene has a ground sampling distance (GSD) of 17×34 cm. Furthermore, the scene referred to as 'Selene H23 Dual' was co-registered from the image data captured by HySpex VNIR-1600 and SWIR-384 sensors. The Dual image has a GSD of 70×70 cm. Photograph of both sensors are shown in figure 3.14.

(a) VNIR-1600 (b) SWIR-384

Figure 3.14: Photograph of the two HySpex hyperspectral cameras from Norsk Elektro Optikk (NEO) company used to capture Selene scene. The photographs are from HySpex website in [73].

The instruments used in this study was collected by DSTL and/or their contractors. The basic sensor properties are listed in table 3.2, which are from HySpex website in [73].

Table 3.2: Properties of HySpex hyperspectral sensors used to capture Selene scenes.

	VNIR-1600	**SWIR-384**
Spectral range	0.41 to 1 µm	0.93 to 2.5 µm
Spatial pixels	1600	384
Channels/bands	160	288
Spectral sampling interval	3.6 nm	5.45 nm
Field of view	17°	16°
Sensor head weight	4.6kg	5.7kg
Radiometric resolution	12 bit	16 bit
Power consumption	30 W	30 W

The HySpex sensors were mounted and flown in south west direction. The recorded positions (as provided by DSTL) are approximately:

	latitude	**longitude**
start	51.14°	−1.65°
end	51.12°	−1.68°

The images from the two sensors were registered together by DSTL and/or contractors to produce the H23 Dual scene. Some of the materials in Selene H23 Dual scene is shown in figure 3.15.

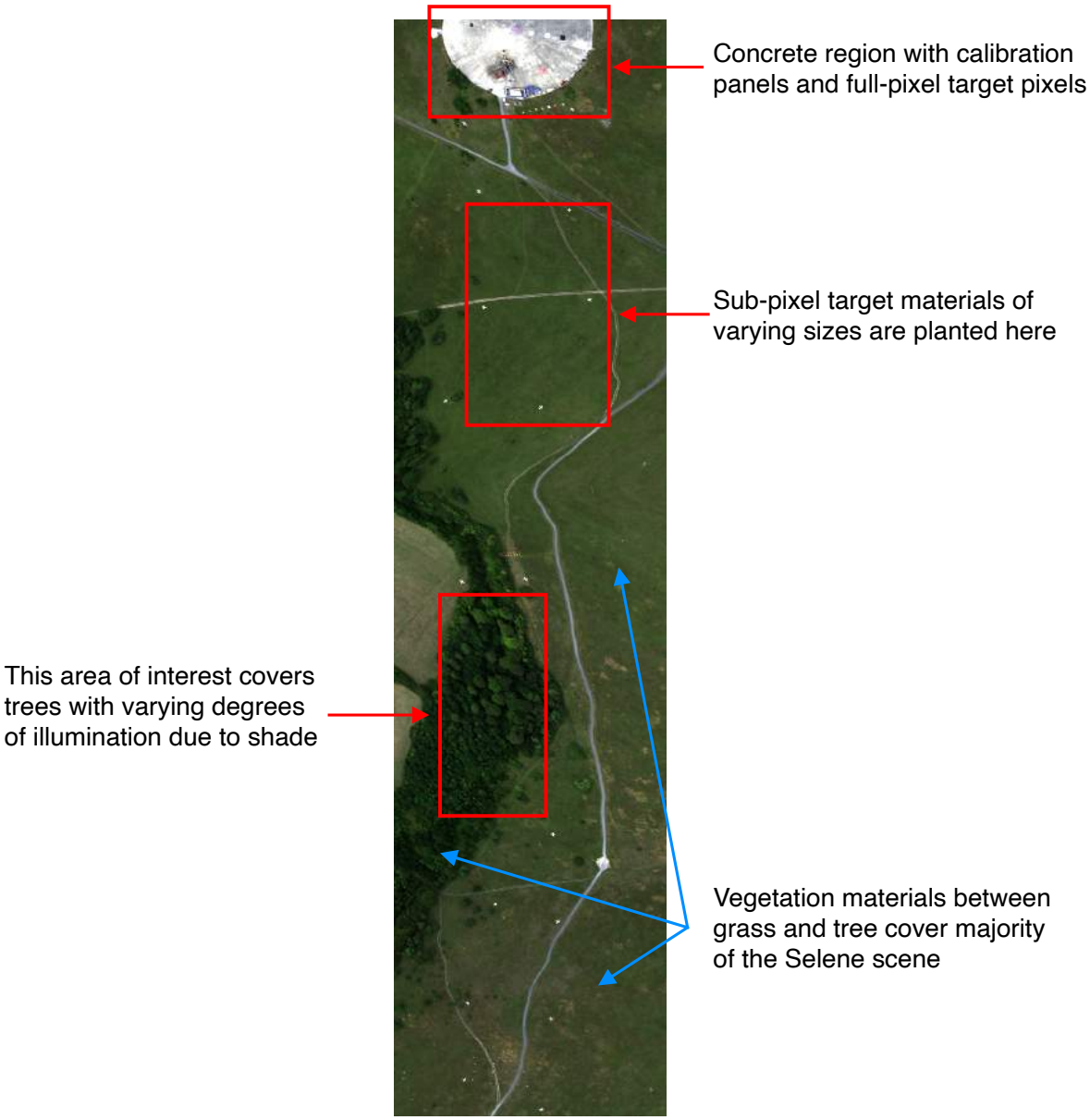
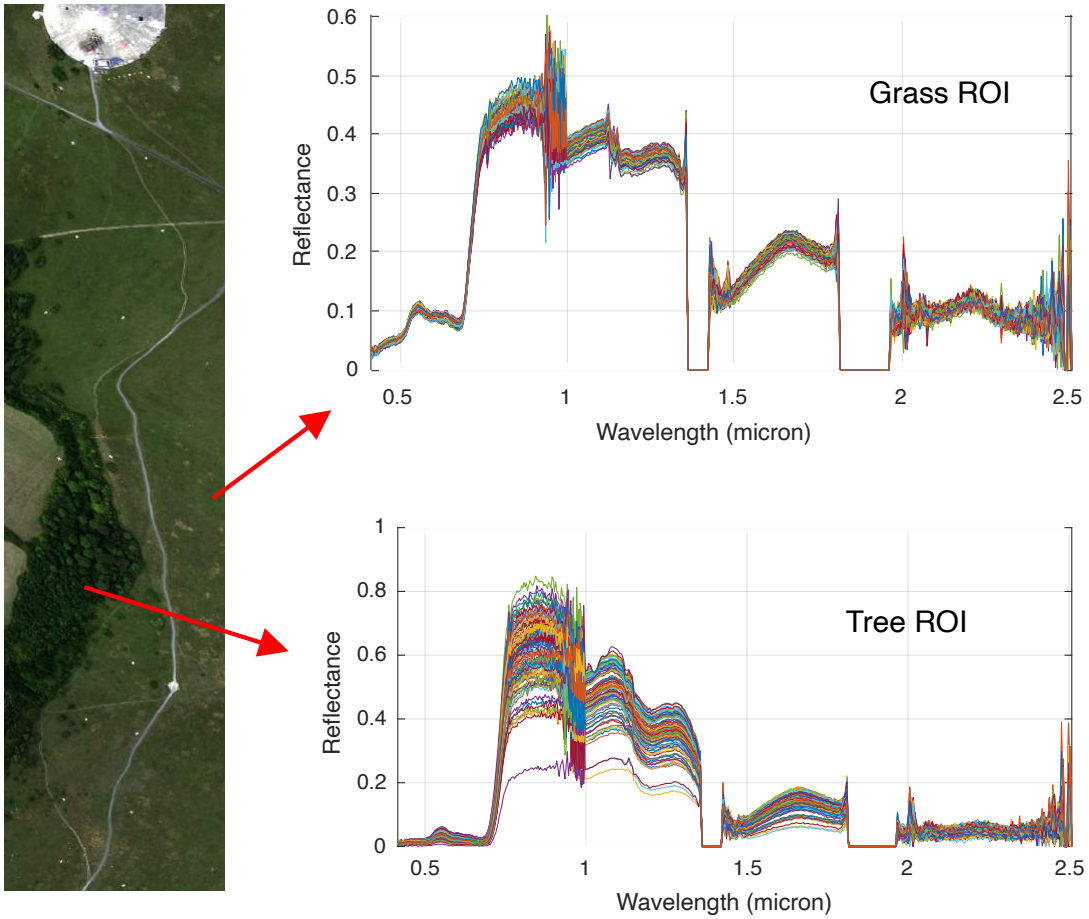


Figure 3.15: Illustrating where different materials are in Selene H23 Dual scene.

Vegetation

The most abundant materials in Selene are vegetation materials. Vegetation materials between grass and tree cover more than 90% of the scene, and natural materials are the ones most susceptible to variability. Selene is covered with a long grass, as seen from a photograph on ground in figure 3.16.

(a)



(b)

Figure 3.16: Depicts the vegetation on Selene H23 with (a) showing a photograph as seen on ground, and (b) showing example spectra of 100 pixels (10 by 10 pixels) of grass and tree in-scene.

Calibration panels

A small portion of the scene contains seven calibration panels placed on the circular concrete. The placement of the panels is photographed in figure 3.17.

Figure 3.17: Photograph of the panels as seen on ground

The purpose of these calibration panels is for radiometric and reflectance calibration of the imaging system. The radiometric calibration is to assess the radiance of the scene, such that the raw data (in digital number) obtained from the sensor can be converted into radiance, which is a more physically meaningful unit. The calibration factor in equation (3.4) given by 'C_F', or ratio between absolute units or radiance 'L' and background corrected sensor counts of the radiometric source 'S_c'.

$$C_F = \frac{L}{S_C} \tag{3.4}$$

To obtain the background corrected sensor counts, the dark current (which is the number of counts detected in the sensor in the absence of light) must be removed from the raw image count. In the Selene dataset, there are a total of seven panels on the concrete for this calibration purpose, and the reflectivity of these panels range from a low 1% (or close to ideal black) to high 99% (as close as possible to an ideal white) as shown in figure 3.18 which depicts the spectral characteristics of these panel in figure 3.19.

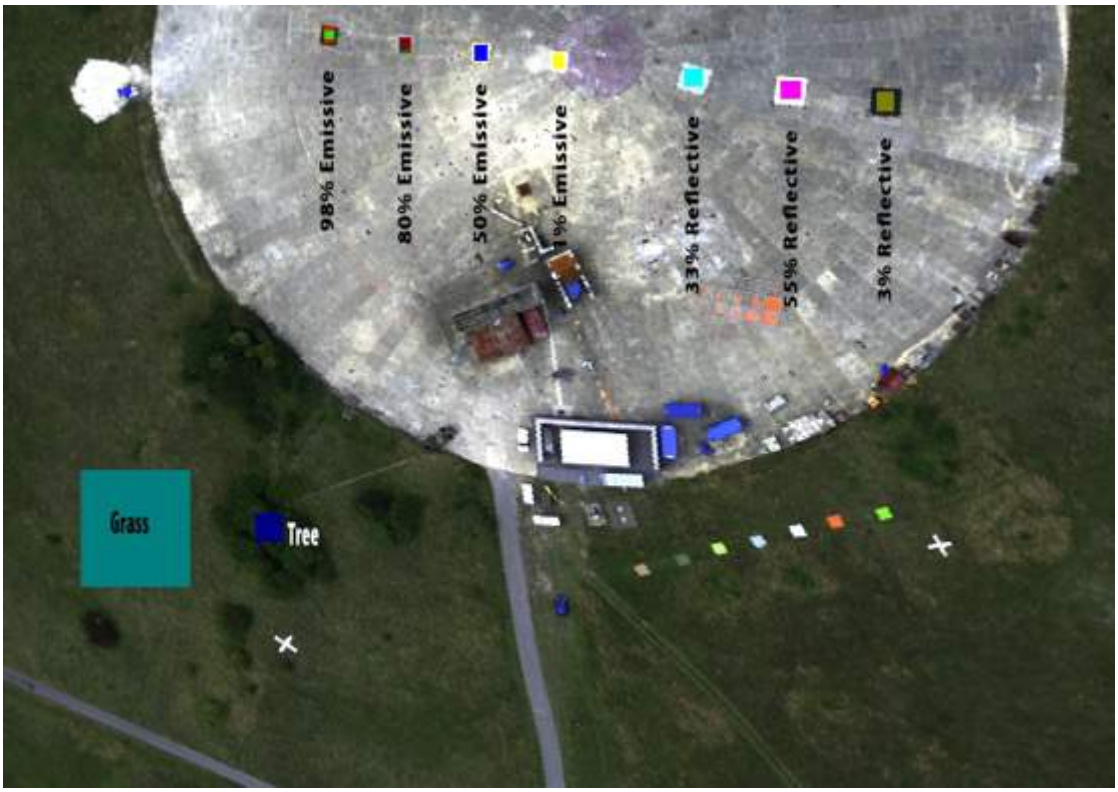


Figure 3.18: RGB image showing the positions of the seven calibration panels ranging from 1% to 99% reflective are placed on concrete material in Selene scene.

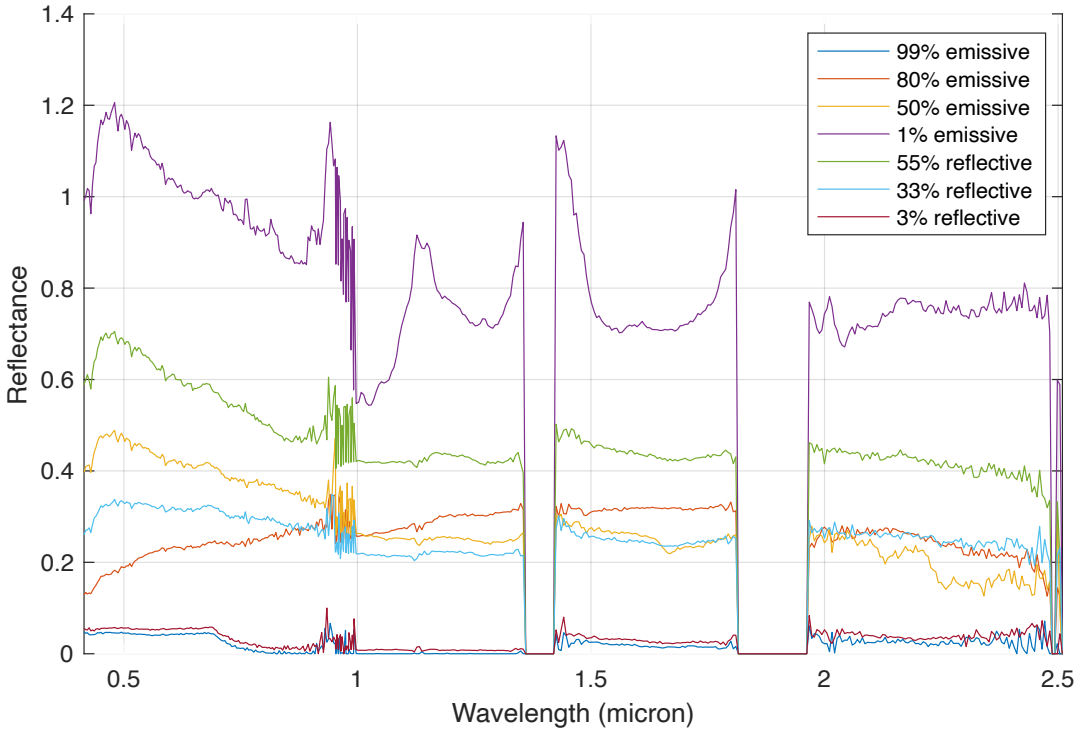


Figure 3.19: Representative spectral information of the panels from a single full pixel of the calibration panels ranging from 1% to 99% reflective are placed on concrete material in Selene scene.

Target materials

The presence of known target materials in Selene makes it the primary dataset of interest in this thesis. Selene has a mixture of full and subpixel target materials scattered throughout the scene, in the same way the calibration panels are presented. With a GSD of 70×70 cm in H23 Dual, the subpixel target materials placed on grass occupy a maximum of ≈2% of a pixel in 10×10 cm targets, and ≈33% in 40×40 cm targets. Orange Perspex is a known target material in the scene, with spectral plot shown in figure 3.20.

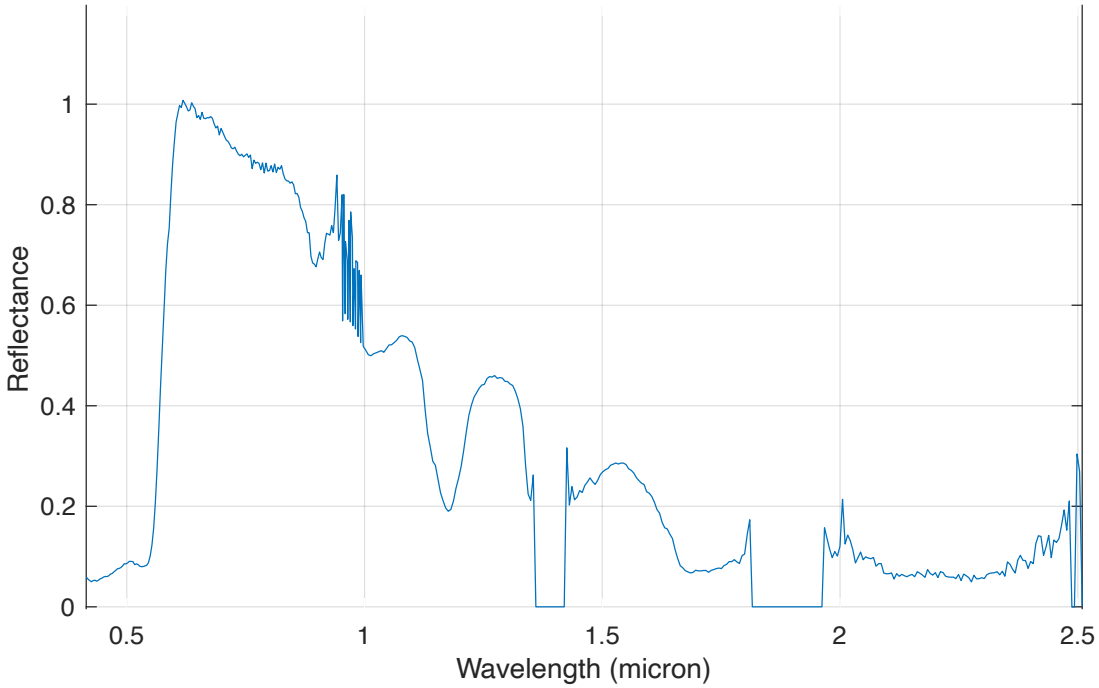


Figure 3.20: An example spectra of Orange Perspex target material.

3.5 Assessment Metrics

3.5.1 Distance Metrics

Two distance measures will be used in this thesis to define the reconstruction accuracy: Manhattan distance and differential ℓ_1 norm error.

Manhattan Distance (MD)

The Manhattan distance (MD) is the sum of absolute point-to-point distance difference between a given ground truth data and the reconstructed data. MD with 'B' bands between two pixels x and x is given by:

$$\text{MD}(x,y) = \ell_1 \text{ norm of residual error } = \sum_{b=1}^{B} |x_b - y_b| \tag{3.5}$$

In this thesis, a robust set of experimental scenes like the scene Selene H23 Dual has 448 bands, whereas, Virginia City scenes have less than half the number of bands than

H23 Dual. So, in situations where the performance comparison is summarised between different scenes, a second table is shown where MD error is normalised by the number of bands in the scene: MD per band.

Differential ℓ_1 norm error (DL1NE)

A second metric used, differential ℓ_1 norm error (DL1NE) demonstrates the difference in area estimates between ground truth and reconstructed samples. While MD shows point to point difference, in order to avoid samples with lower absolute value in magnitude being overshadowed by samples with higher magnitude, DL1NE measured is shown in percentage in this thesis. For example: a 200% DL1NE error estimate suggests that the reconstructed pixel (x) has three times the ℓ_1 norm value for the same ground truth pixel (x). DL1NE is given by:

$$\text{DL1NE}(y,x) = |\frac{||y||_1 - ||x||_1}{||y||_1}| \times 100 \tag{3.6}$$

3.5.2 Detection metrics for minority materials

Anomaly detection with RX

Spectral distance error metric gives an indication of the goodness of the reconstruction only for the background materials which are abundant in the scene. However, these methods are not sufficiently sensitive enough to quantify the errors of very trace materials, e.g. the minimal number of artificial materials typically few % in the natural scene (e.g. the colour panels in the Selene data set). Reed-Xeoli (RX) anomaly detection [74] is one way to identify small targets relative to the background. RX algorithm is formed under the Generalized Likelihood Ratio Test (GLRT) framework for multidimensional image data. RX algorithm assumes that the spectrum of the target and the covariance of the background are Gaussian distributed and that they are generally unknown. RX is shown in equation (3.7) for ith pixel where x_i is the ith pixel in question, $\bar{x}$ the mean of the whole scene, C is the inverse covariance matrix.

$$\mathrm{RX}_i = (x_i - \bar{x})^T C (x_i - \bar{x}) \tag{3.7}$$

The RX of the reconstructed scene compared to the original ground truth data is evaluated, in this thesis, it is achieved through the Pearson correlation (PC) coefficient. The cosine similarity between two non-zero vectors 'a' and 'b' measures the cosine of the angle between them. A PC score is meant to deduct the similarity within -1 to +1 for negative and positive correlations. A PC score of +1 signifies that the reconstructed data has a very strong positive correlation with ground truth RX, and a score of -1 suggests a strong negative correlation. Equation (3.8) shows the PC between two vectors 'a' and 'b'.

$$\mathrm{PC}(a,b) = \left(\frac{a-\bar{a}}{||a-\bar{a}||_2}\right)^T \left(\frac{b-\bar{b}}{||b-\bar{b}||_2}\right) \tag{3.8}$$

Target detection with ACE

In the case where a target material is known, the target detection algorithm known as the adaptive cosine estimator (ACE) [75, 76] has been adopted for testing the ability of the reconstruction to recover trace targets in the scene. ACE is one of the classical target detection (TD) methods whose individual distance measures is the square of the Mahalanobis distance. ACE is shown in equation (3.9) for *i*th pixel where s is the known target, x_i being the *i*th pixel in question, $\bar{x}$ the mean of the whole scene, C is the inverse covariance matrix.

$$\mathrm{ACE}_i = \frac{((s-\bar{x})^T C (x_i - \bar{x}))^2}{((s-\bar{x})^T C (s-\bar{x}))((x_i-\bar{x})^T C (x_i-\bar{x}))} \tag{3.9}$$

The detectability or the extent of which target material is detected in the midst of background is presented by means of a receiver operating characteristics (ROC) curve. The ROC curve is a graphical plot which plots the probability of detection (PD) against the probability of false alarm (PFA) for the detection of specific target. PD is the number

of true positives divided by the total number of target pixels at a given point. And, PFA is the number of false positives against the total number of background pixels at a given point. An example of the ROC is illustrated in figure 3.21. The locations of targets (i.e. the target map) is noted from the ground truth data set (i.e. the Selene ground truth data in this paper). The ROC of the target is constructed by using the scores given by ACE detection of the target in the reconstructed scene, and it is then constructed using the target map information to indicate the faithfulness of the detection. The 'goodness' of the detection is evaluated with the area under curve (AUC), which is numerical index used to describe the behaviour of the ROC curve. The AUC for target 't' is given by:

$$\mathrm{AUC(t)} = \int_0^1 \mathrm{ROC(t)}\delta t \tag{3.10}$$

An AUC with a value of '1' shows perfect classification of target material without false positives and an AUC of '0' shows that none of the target pixels are accurately classified.

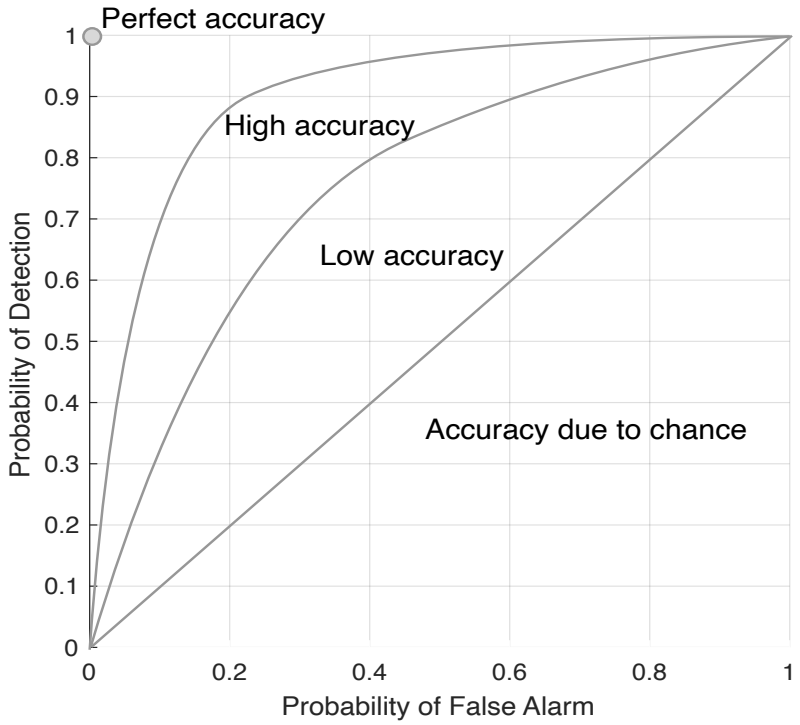


Figure 3.21: The figure illustrates the implication of the receiver operating characteristics (ROC) in target detection. [3]

3.6 Chapter summary

Due to the high investment capital required to setup hyperspectral cameras (with some hyperspectral camera setups costing in order of a few hundred thousand pounds), scene simulation is a time and cost saving alternative to gathering field data through an experimental trial. CameoSim scene simulator is a commercial-off-the-shelf (COTS) simulator used by defence agencies worldwide for the generation/simulation of multispectral and hyperspectral image data.

COTS simulators constraint a maximum of four material per pixel during the process of scene reconstruction. When the spatial resolution of the image is degraded, for example when the images are taken from high altitude, the actual number of material species contain in these badly resolved pixel can be more than four. In this case the multiple endmember (EM) in a pixel is required to be approximated into the limits of the model (i.e. 4) which increases the error of the reconstruction.

Finally, this chapter provides details of the experimental setup which include six hyperspectral images, including a scene called 'Selene' with known subpixel targets. The images are evaluated by their mean point to point difference or Manhattan distance with ground truth data, and differential L1 norm error (DL1NE) to estimate the error difference in terms of the area. Trace and target materials are evaluated by RX and ACE detection algorithms.

Chapter 4

Material Allocation and its Impact on Spectral Reconstruction

This chapter presents an understanding behind material allocation, the methods used by CameoSim, and it is potential spectral reconstruction impact from multispectral inputs. The understanding for spectral reconstruction from a pre-trained library is, given the appropriate multispectral input bands, the algorithm will be able to generate a representation for the test dataset with the same accuracy as hyperspectral input. The appropriateness of the input bands is subjective to the training and test dataset, including the noise levels of each. The results presented in this chapter works with reflectance of MSI and HSI data, and scene simulation with MSI input is performed only in Chapter 6.

This chapter is structured by first providing the principle behind spectral reconstruction in section 4.1 and details of the material allocation methods currently in use in CameoSim like commercial scene simulators in section 4.2. This is followed by evaluation of reconstruction from existing spectral similarity measures in section 4.3.2 and distance measures in section 4.3.1. These studies are followed by an explored direction of reconstruction porting Matrix R pseudoinverse method onto reflectance data in section 4.4. This chapter concludes with section 4.5 covering the recent advances of spectral reconstruction from dictionary learning unmixing methods.

4.1 Principle behind spectral reconstruction

Hyperspectral Imaging (HSI) provides detailed spatial and spectral information fundamentally crucial for many applications such as earth observations, surveillance and spectroscopic analysis, etcetera. However, due to the relatively more expensive instrumentation in the HSI system, detailed hyperspectral data is not widely available. Many satellite imageries, for example, the WorldView and Landsat, etcetera., capture multispectral imageries (MSI) of the scene in less than ten spectral bands. Spectral reconstruction is a principled approach to convert MSI data to HSI.

There are two general approaches to achieve SR: the direct method which requires the characteristics of the camera system and the indirect reconstruction, which is also known as the learning-based method. Research for the latter approach, such as the matrix R method [77][78], reconstruction as an inverse problem [79], and dictionary learning (DL) [80] have been actively pursued in the past decade. Matrix R and its variant, such as the regularisation version, have been shown to reconstruct HSI within $<10\%$ error effectively. Mathematically, this inverse problem of SR can be represented in equation (4.1) for a pre-trained dictionary 'D'.

$$\min ||y||_{0,\text{bands}} \text{ from } y = Da \text{ s.t. } a_{\text{MSI}} = a_{\text{HSI}} \tag{4.1}$$

4.2 Prior work in material allocation

To recall, the texture module in CameoSim is where EM-abundance is given as input in LUT file with a maximum limit of four materials per pixel. While this is the generic method to input material allocation, CameoSim like other commercial HSI simulators has a few in-built methods implemented that make this unmixing process simpler for the simulator users. Methods such as nearest matched material and texture material mapper (TMM) are the methods that have been employed by the off-the-shelf simulators for matching the EM materials from a pre-compiled material library (a database of

endmember materials compiled from publicly available sources).

4.2.1 Experimental Setup

Material Library or Spectral library used in this chapter is compiled from several publicly available sources, namely from the NASA's Jet Propulsion Laboratory with libraries such as ECOSTRESS Spectral Library (accessed from the web URL `https://ecostress.jpl.nasa.gov`) version 1, formally known as ASTER, and HyspIRI Ecosystem Spectral Library version 1 (accessed from the URL `https://hesl.jpl.nasa.gov`), USGS Spectral Library version 7, Spectral Library of Impervious Urban Materials (SLUM) (accessed from the URL `http://www.met.reading.ac.uk/micromet/LUMA/SLUM.html`), and over 20 minor libraries compiled by universities across USA and UK.

Spectral reconstruction is performed with 8 centre wavelengths of the WorldView-2 (WV2) satellite and the 16 Wavelengths of the WorldView-3 (WV3) satellite. A one-dimensional plot of the location of the centre wavelengths of both the WV2 and Wv3 is shown in figure 4.1.

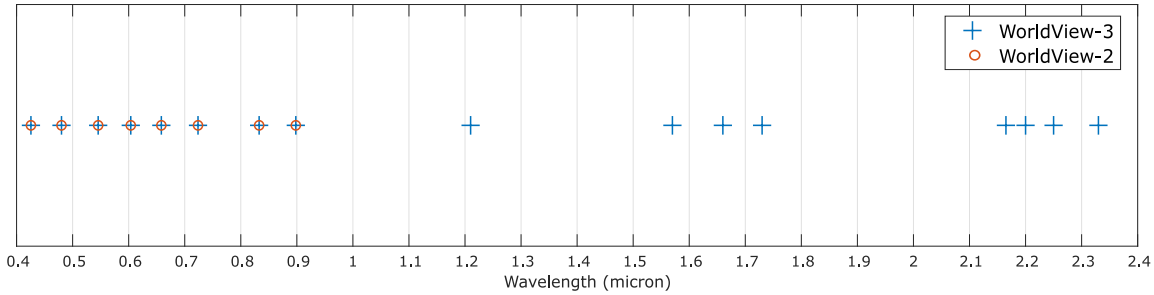


Figure 4.1: One-dimensional plot to show the centre wavelengths of WorldView-2 with 8 spectral bands from 0.4 to 0.9 µm and WorldView-3 (WV3) with 16 spectral bands from 0.4 to 2.4 µm used in this thesis.

It is understandable that there will not be an exact match between the centre wavelengths of the MSI satellites, the HSI images, and the compiled material library. A given library contains hyperspectral resolution (100s of bands). In such cases, cubic spline interpolation is used for a smoother curve to subset the HSI image to the wavelengths of the MSI satellites. The abundance is estimated by the material allocation

methods, and the pixel is replaced with the selected library material with the estimated abundance. Thus, the spectral reconstruction process is described in three steps:

- The first step is to subset the library dictionary ($D_s \subset D$)
- The second step is to estimate the representation/abundance 'a' for signal sample/pixel with MSI resolution 'y_{MSI}' by solving the objective $arg\min_a(y_{\text{MSI}} - D_s a)$
- The final step is to reconstruct the HSI resolution for 'y' by ($D \times a$)

4.2.2 Closest matched material

Closest or nearest matched material replaces each pixel with the library material with the closest Euclidean distance. This method considers all pixels to be a pure pixel, and when applied supervised with a pre-compiled material library can match inaccurately for a mixed pixel. The pseudo-code for closest matched material is given in algorithm 8.

Algorithm 8 The closest matched material model for material allocation

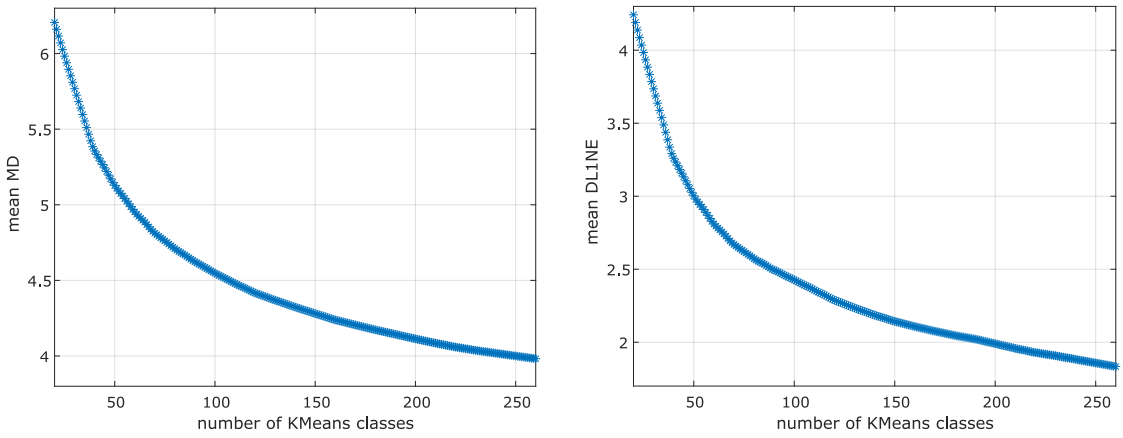
1: **Input** pre-compiled EM dictionary or material library 'D', pixel 'y'
2: **Estimate** Euclidean distance ED(D, y) of all EM dictionary elements w.r.t. y.
3: **Let** position of minimum ED error as 'p'
4: **Output** the position 'p' and abundance as '1' for the selected EM materials

The closest matched material method is comparable to an equivalent unsupervised application, which is with k-means classification material allocation method, where each pixel is substituted by its closest cluster centres. KMeans or Lloyd's algorithm iterates between cluster centres (C) and membership of a pixel Ys (Ys $\subset$ full image Y) at each iteration, where each pixel is allowed only one membership. The cluster centre (Ci $\subset$ C) is known as centroid if the centre is the mean value of the all the members and called clustroid if it is the value of a class member closest to the mean.

In this type of approach for material allocation, each pixel Ys selects the spectrum of its cluster centre Ci closest to its value. The image is reconstructed by substituting/directly replacing the value of the pixel with the value of its Ci, which is performed in two steps:

- Step 1 is to run KMeans algorithm to find the cluster centres Ci and the membership of each pixel,
- Step 2 is to directly replace the spectrum of each pixel with the value of Ci.

Application of this method for the 448 bands of Selene H23 Dual scene, as shown in figure 4.2, which shows a best-case error of ≈1.8% DL1NE when 256 classes (the maximum number of EM materials allowed in the current version of CameoSim) of materials are employed for the scene simulation. The high error is because with unsupervised classification approach like the KMeans as closest matched material option, a single pixel is only allowed to be part of 1 cluster centre and the concept of partial/fractional memberships to include subpixel materials is absent.



(a) (b)

Figure 4.2: The decrease in the (a) MD error, and (b) DL1NE for the closest matched material method with increase in KMeans for classification on Selene H23 Dual scene.

Figure 4.3 shows the DL1NE error false colour map when the simulation employs 80 k-means clusters for the simulation, and it results with a mean error of 2.57%.

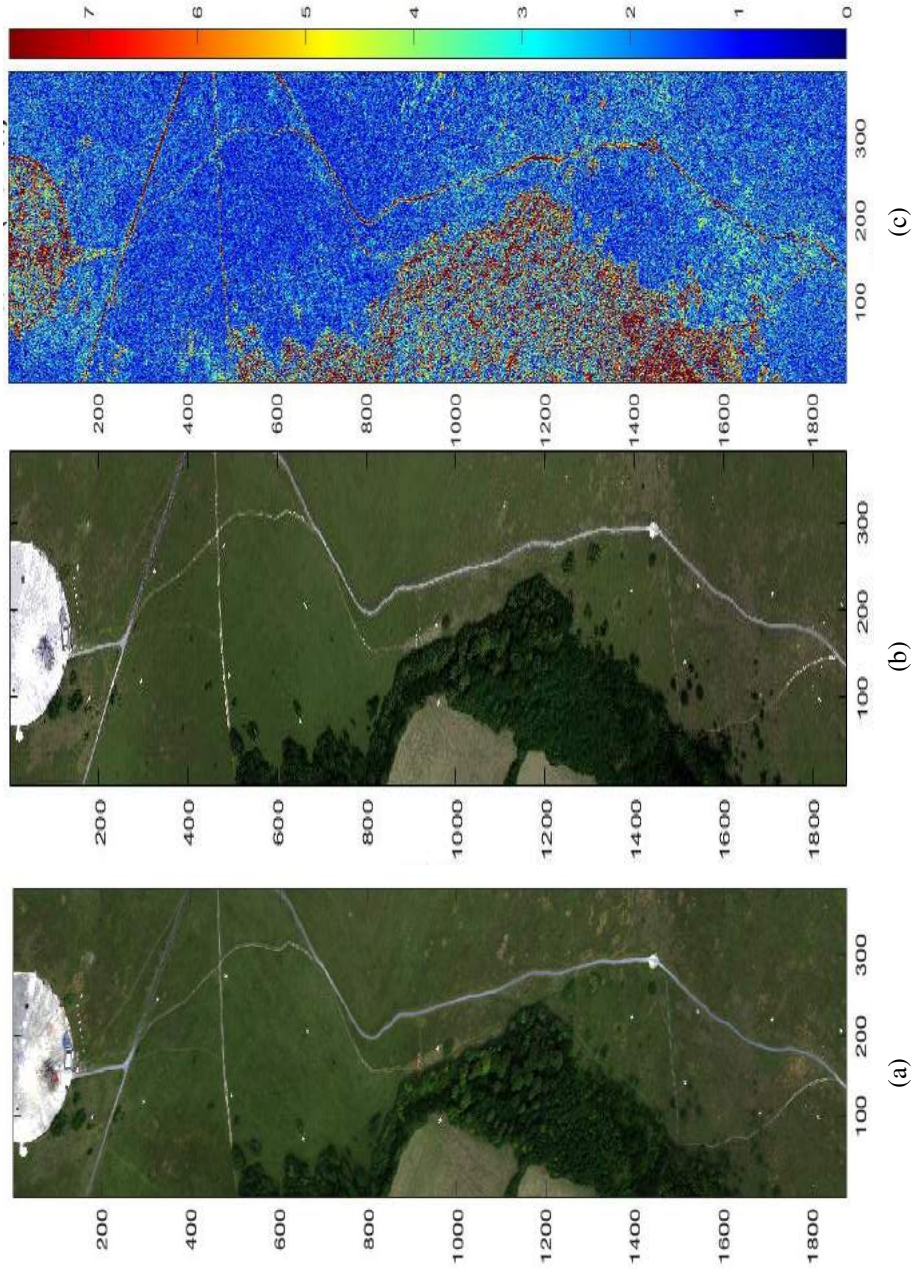


Figure 4.3: Closest matched material with 80 classes k-means clustering on all bands of H23 Dual image. The sub-figures (a) and (b) shows the RGB colour image of the ground truth and reconstructed scene respectively, and (c) shows the DL1NE error map with mean error of 2.57%. The colorbar shown is upper bounded by 3x mean for visual purposes.

4.2.3 Texture Material Mapper (TMM)

Material mapper is the default method in CameoSim for material allocation. The texture material mapper (TMM) technique [81] has been used extensively in the HSI simulator (e.g., the CameoSim package) [82, 83, 84], which estimates the abundances by evaluating the inverse of the Euclidean distances of each EM with respect to the spectral characteristics of the mixed pixel, i.e., the test pixel.

TMM is a two step process for a single pixel. Given EM dictionary 'D' and a pixel sample, the first step of the TMM is material selection. For a mixed pixel 'y', the first N_{mp} materials from the 'D' with the lowest Euclidean distance (which is the sum of squared error) is selected. Once the EM materials are selected from the 'D', the next step is to estimate the abundance of a material, which is calculated as the inverse of the distance and then normalised for the sum to unity. The TMM abundance for N_{mp} number of materials and selected EM dictionary at position 'i' is given by:

$$\text{TMM}(D_i, y) = \frac{a_i}{\sum_{j=1}^{N_{mp}} a_j}, \text{ where } a_i = \frac{1}{ED(D_i, y)} \tag{4.2}$$

Algorithm 9 shows the TMM abundance of material within the pixel. The TMM was run with the compiled material library and was found with a high reconstruction error of a mean 7.81% DL1NE with the centre wavelengths of WorldView-3 (WV3) as input, and a mean of 20.54% with WorldView-2 (WV2) input. The results are presented in figure 4.4.

Algorithm 9 Estimating texture material mapper (TMM) abundance

1: **Input** pre-compiled EM dictionary or material library 'D', pixel 'y', and number of materials per pixel as 'N_{mp}'
2: **Estimate** Euclidean distance ED(D, y) of all EM dictionary elements w.r.t. y.
3: **Sort** and select the 'N_{mp}' EM materials with the smallest values in ED(D, y)
4: **Estimate** abundance 'a' for the N_{mp} selected EM materials from equation (4.2)
5: **Output** the position and abundance of N_{mp} selected EM materials

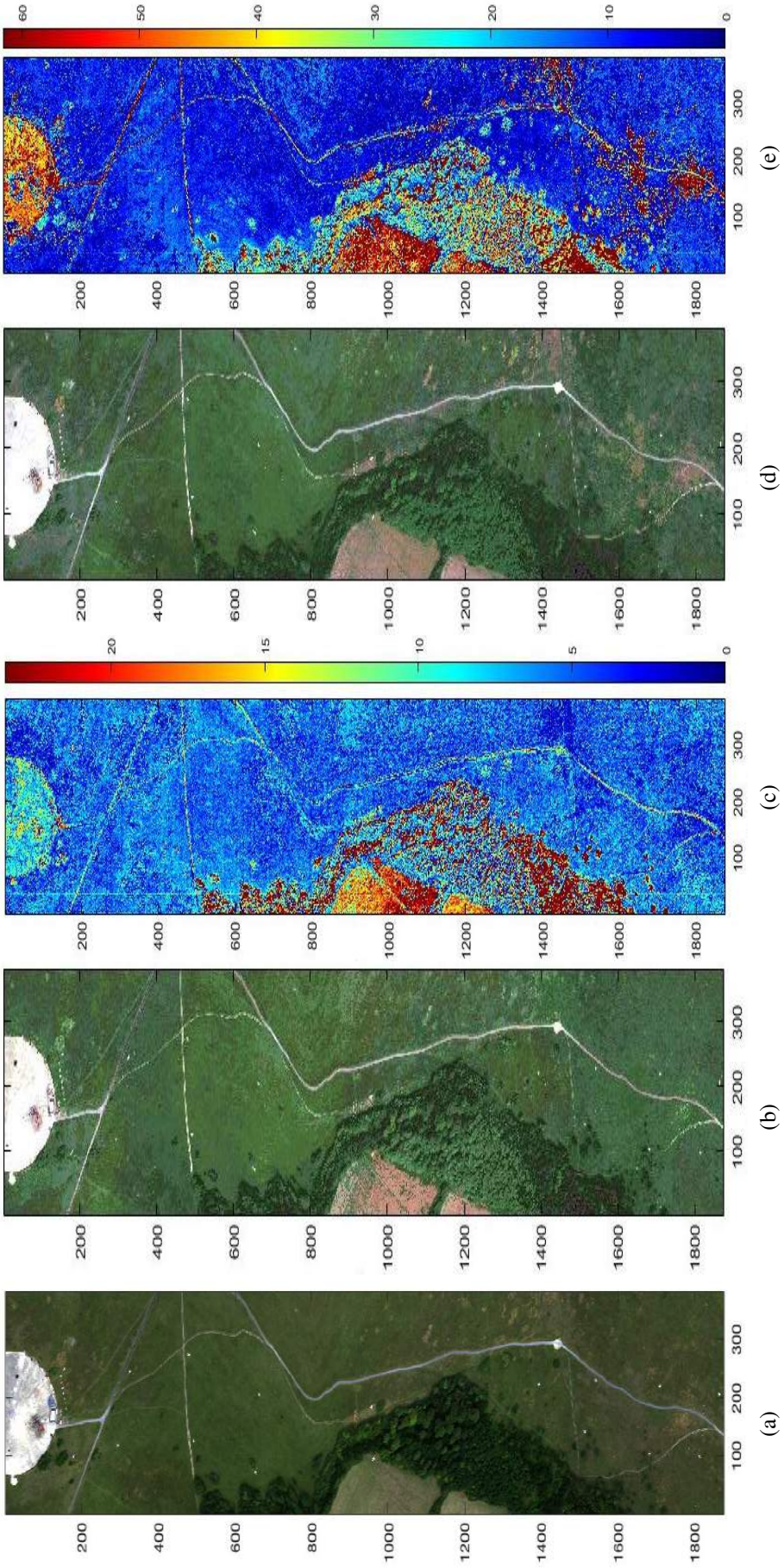


Figure 4.4: TMM material mapper for spectral reconstruction on Selene H23 Dual scene. The sub-figures (a) show the ground truth RGB, (b) and (c) the RGB and DL1NE error map of the reconstructed scene from WV3 input, and (d) and (e) are the RGB and DL1NE error map of the reconstructed scene from WV2 input. Figures (c) and (e) are with a mean of 7.81% and 20.54% respectively. The colorbar shown is upper bounded by 3x mean for visual purposes.

4.3 Exploration of potential Material Mapper (MM) techniques using distance and similarity metrics

This section evaluates existing distance measures in subsection 4.3.1 and spectral spectral similarity measures in subsection 4.3.2 that can be potentially used instead of the TMM method, and evaluates the spectral reconstruction accuracy taking in the centre wavelengths of WorldView-2 (WV2) and WorldView-3 (WV3) satellite image as input.

4.3.1 Distance measures

The general mathematical equation of ℓ_p norm for a total of 'i' dimensions is given by:

$$\ell_p \text{ of vector 'x' } = (\sum_i |x_i|^p)^{\frac{1}{p}} \tag{4.3}$$

The Manhattan distance (MD) is when p is 1, or sum to absolute distance difference. Euclidean Distance (ED) on the other hand is when p = 2, mathematically represented as:

$$\text{ED}(x) = (\sum_i x_i^2)^{0.5} \tag{4.4}$$

Over the years, ED has been portrayed in several ways for selection of materials, including the Normalized Euclidean Distance (NED), mentioned in [85]. NED is where both the EM dictionary elements 'D_i' is first normalised along with the pixel/signal samples 'y', and ED is estimated between the normalised vectors. The normalisation factor is often referred to as the area under curve. While dividing a signal by the area under the curve does eliminate illumination artefacts and shade of a colour present in the scene [86], these errors are mostly from the drawbacks in an atmospheric correction algorithm's inability to perfectly mask the shadowed areas. However, a great disadvantage is that the a method like NED loses its absolute value of reflectance.

4.3.2 Spectral similarity measures

Spectral angle measurements like Spectral Angle Mapper (SAM) and Spectral Correlation Angle (SCA) [85] are more concerned with the similarity in the shape between two vectors. The equation of SAM is written as:

$$\mathrm{SAM}(x,y) = \cos^{-1}\left(\left(\frac{x^T}{||x||_2}\right)\left(\frac{y}{||y||_2}\right)\right) \tag{4.5}$$

Spectral Gradient Angle (SGA) is another class of methods that builds spectral angle on the first derivative or gradient of each of the vectors. Similar to spectral angle measures, spectral information measures like Spectral Information Divergence (SID, equation (4.6) for 'B' bands).

$$SID(x,y) = \sum_{b=1}^{B} \hat{x}_b \log\frac{\hat{x}_b}{\hat{y}_b} + \sum_{b=1}^{B} \hat{y}_b \log\frac{\hat{y}_b}{\hat{x}_b}, \text{ where } \hat{x} = \frac{x}{||x||_1} \text{ and } \hat{y} = \frac{y}{||y||_1} \tag{4.6}$$

And, hybrids combining two difference measures like SIDSIN and SIDTAN [87] which are combinations of SID and SAM are all designed to predict the similarity of the target spectra in question with individual pixels on ground. SIDTAN and SIDSIN are shown in equations (4.7) and (4.8) respectively.

$$SIDTAN(x,y) = SID(x,y)\tan(SAM(x,y)) \tag{4.7}$$

$$SIDSIN(x,y) = SID(x,y)\sin(SAM(x,y)) \tag{4.8}$$

Apart from statistical correlation like Root Mean Square Error (RMSE) and R^2 statistic, other methods like Spectral Correlation Angle (SCA, equation (4.9b)) builds on

the spectral angle for the correlation between the EM dictionary and signal sample, have also been attempted. The spectral correlation (SC) or the Pearson correlation coefficient introduced in the previous chapter estimates the correlation between two vectors which range between -1 and +1 for strong negative and positive correlation respectively. An addition of +1 to the SC removes negative range from the numerator and the SCA estimates the angle with a range between 0 and $\frac{\pi}{2}$. Other techniques such as the Spectral Gradient Angle (SGA) which builds spectral angle on the gradient or first derivative of the vectors, and combination of other methods such as the Normalized Spectral Similarity Search (NS^3, [88]) are reported.

$$\mathrm{SC}(x,y) = \left(\frac{(x-\bar{x})}{||x-\bar{x}||_2}\right)^T \left(\frac{(y-\bar{y})}{||y-\bar{y}||_2}\right) \tag{4.9a}$$

$$\mathrm{SCA}(x,y) = \cos^{-1}\left(\frac{SC(x,y)+1}{2}\right) \tag{4.9b}$$

While all these similarity measures fitting provide spectral angle minimisation for the given input wavelengths, it does not account for the differences in absolute distance that occurs in extrapolated data points. For extrapolation of data, such manipulation of magnitude of vectors results in pixels having reflectance greater than 1 or less than 0 at some wavelengths. Figures 4.5 and 4.6 show the reconstruction accuracy from WV2 and WV3 centre wavelength MSI input, and tables 4.1 and 4.2 show a summary of the common spectral similarity and distance measures when used for material selection.

Table 4.1: DL1NE error for spectral reconstruction with MSI input on Selene H23 Dual scene using a pre-compiled material library

MM Method	WV3 input	WV2 input	Mean
ED (TMM)	7.81	20.54	14.18
NS3	10.2	19.21	14.71
SAM	**6.44**	**12.08**	**9.26**
SID	6.48	12.46	9.47

Table 4.2: MD error for spectral reconstruction with MSI input on Selene H23 Dual scene using a pre-compiled material library

MM Method	WV3 input	WV2 input	Mean
ED (TMM)	14.02	20.95	17.49
NS3	14.41	20.41	17.41
SAM	**12.26**	16.87	**14.57**
SID	12.50	**16.73**	14.62

The tables summarise that SAM and SID are suitable alternatives to the TMM algorithm using ED with an increased accuracy of ≈53% in DL1NE and ≈16.7% in MD using SAM, and ≈33% in DL1NE and ≈16.4% in MD using SID. The methods, however, produce ≈10% DL1NE in reconstruction error in material allocation, and further avenues have been explored in the thesis that produce a lower error.

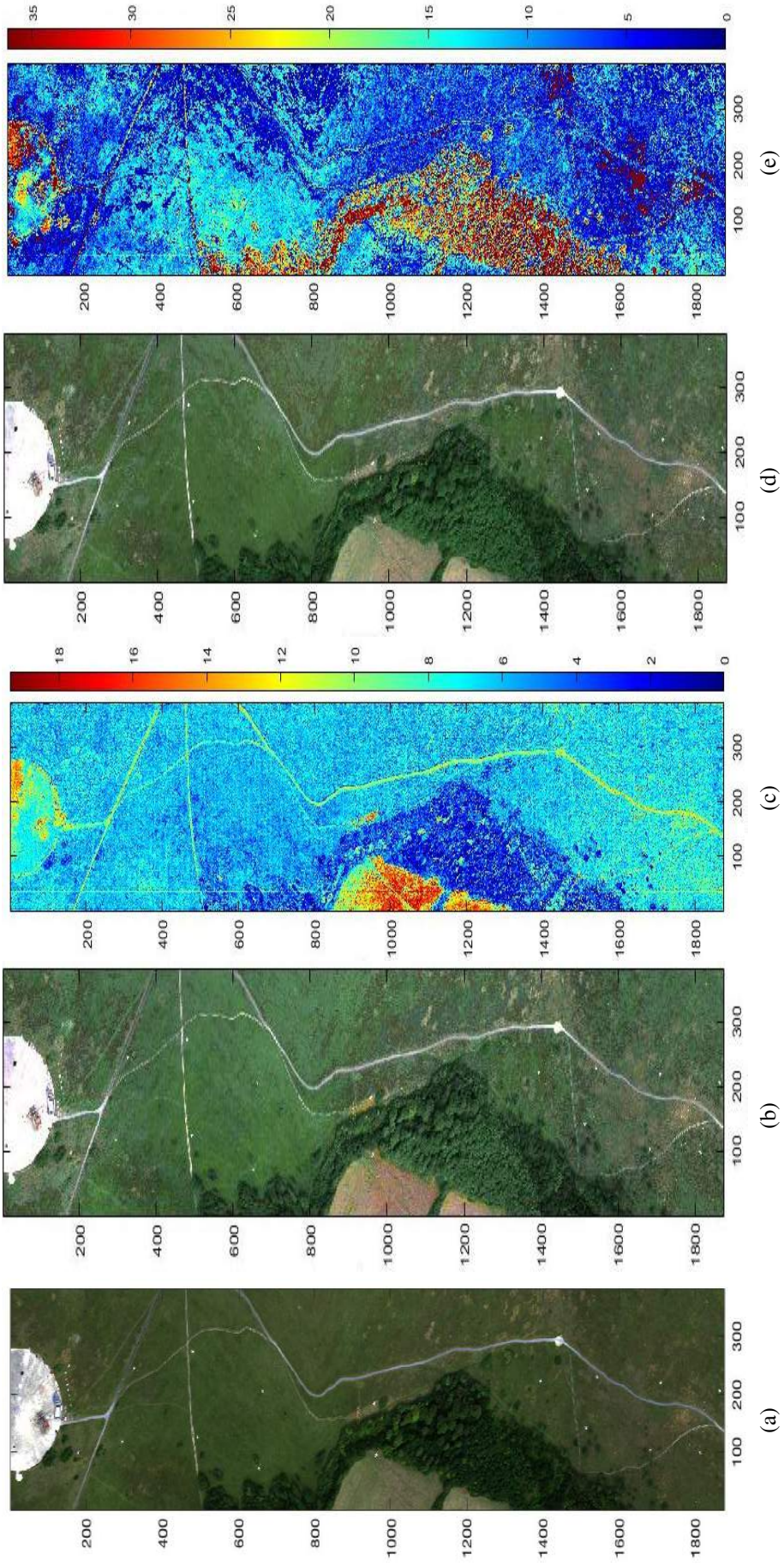


Figure 4.5: SAM for selection of materials in material mapper for spectral reconstruction on Selene H23 Dual scene. The sub-figures (a) show the ground truth RGB, (b) and (c) the RGB and DL1NE error map of the reconstructed scene from WV3 input, and (d) and (e) are the RGB and DL1NE error map of the reconstructed scene from WV2 input. Figures (c) and (e) are with a mean of 6.44% and 12.08% respectively. The colorbar shown is upper bounded by 3x mean for visual purposes.

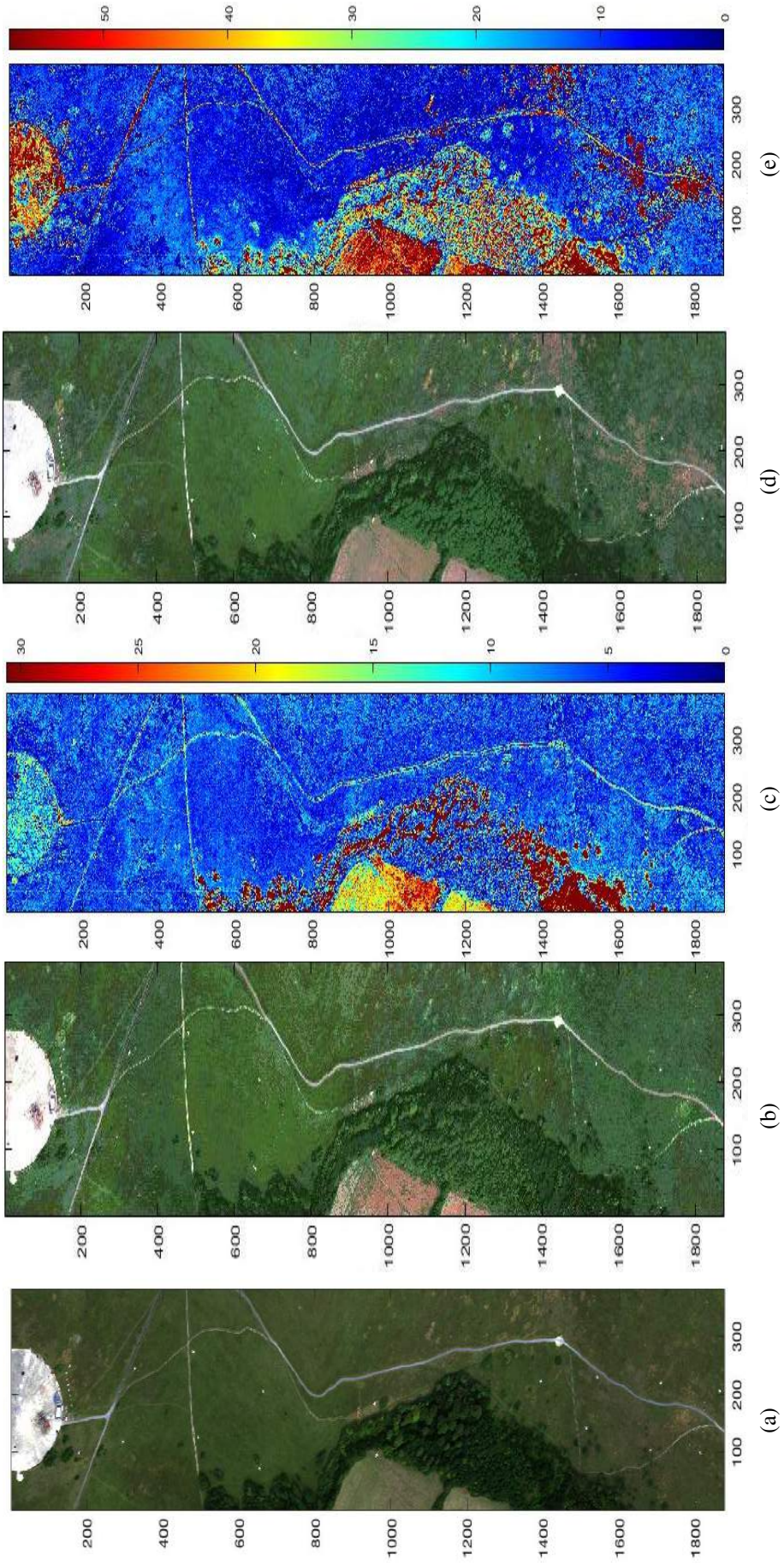


Figure 4.6: NS3 for selection of materials in material mapper for spectral reconstruction on Selene H23 Dual scene. The sub-figures (a) show the ground truth RGB, (b) and (c) the RGB and DL1NE error map of the reconstructed scene from WV3 input, and (d) and (e) are the RGB and DL1NE error map of the reconstructed scene from WV2 input. Figures (c) and (e) are with a mean of 10.2% and 19.21% respectively. The colorbar shown is upper bounded by 3x mean for visual purposes.

4.4 Spectral reconstruction adopting Matrix R method

4.4.1 Theory of Matrix R

Matrix R[89], theorised by Wyszecki, state that the energy distribution of any colour stimulus can be decomposed into two properties: the fundamental stimulus 'N' and the metameric black 'B'. The stimulus 'N' can be the spectral reflectance, transmittance of the surface, the spectral radiance or the irradiance of the illumination source. The fundamental stimulus is the orthogonal projection of the stimulus N through a matrix R, which, is constructed by a colour matching function matrix A as follows:

$$A=ksv \tag{4.10}$$

where k is the constant scalar adjustment factor of the tristimulus values of the three primaries, s is the light spectral power distribution function (which is 3 by 'p' matrix where 'p' is the number of data points sampled), and v is the standard observer matching function (is a 'p' by 1 vector). Matrix R is defined as:

$$R=A(A^TA)^{-1}A^T \tag{4.11}$$

and thus the fundamental stimulus N*:

$$N^*=RN \tag{4.12}$$

and the metameric black B:

$$B=N-N^* \tag{4.13}$$

The R matrix can be obtained through experiment by setting up a series of calibration data and therefore this reconstruction method is only applicable for data sets which are acquired through well-planned experimental trials. Spectral reconstruction of other data sets, such as those of the satellite imagery, will need alternative techniques. Furthermore, the MR method for SR application has only been validated for the visible

spectral range and its applicability for other spectral regions, such as the SWIR bands, remains to be a subject of further research.

4.4.2 Extension of Matrix R to HSI

Principle of algorithm

Colour systems for example the CIELAB or CIEXYZ standards, encrypts the scene into the form like that shown in equation (4.14) and the signal u_i is integrated over visible wavelengths (λ).

$$u_i = \int \overbrace{E(\lambda)}^{\text{irradiance}} \underbrace{r(\lambda)}_{\text{reflectance}} \overbrace{t_i(\lambda)}^{\text{CIE functions}} \partial\lambda \tag{4.14}$$

The $t_i(\lambda)$ in the above equation is a function of spectral transmittance of optical system, spectral transmittance related to *i*th filter, spectral sensitivity of the camera, and noise. By putting $S = E(\lambda)t(\lambda)$ then equation (4.14) in matrix notation can be expressed as:

$$u = Sr \tag{4.15}$$

Solving $f(S|u,r)$ in equation 4.15 results in a 3 by n matrix formed by a weighting table under a particular illuminant and observer, 'n' being the number of output bands in the visible wavelength region. Note that this equation is very similar to 4.12 and the transformation matrix R in 4.12 is similar to the S in 4.15, therefore the pseudoinverse method (equation 4.16) may be a feasible way to obtain the stimulus if a set of calibration data for training is available:

$$u_{train} = Sr_{\text{train}}$$

$$S = u_{\text{train}} r_{\text{train}}^{-1} \approx u_{\text{train}} \overset{\dagger}{r}_{\text{train}} \tag{4.16a}$$

$$u_{\text{test}} = Sr_{\text{test}}$$

$$u_{\text{test}} \approx (u_{\text{train}} \overset{\dagger}{r}_{\text{train}}) r_{\text{test}}, \forall \{u,r\}_{\text{train}} \supseteq \{u,r\}_{\text{test}} \tag{4.16b}$$

Rapid Reconstruction with Pseudoinverse Algorithm

The central limit theorem states that the probability distribution of the average of independent random variables (reflectance of materials in this case) converge to a normal distribution as the number of observations increases. The explored direction employs the source separation model in the context of pseudoinverse method covered in Matrix R section of the thesis, to quickly reconstruct a few highly correlated wavelengths.

Reflectance of a material is due to the chemical/atomic structure of the material and thus the wavelength dependence of the reflectance is the result of the electronic structure of the material. The reflectance is the result of electronic excitation/de-excitation of electron states of material, there is no obvious statistics to describe the wavelength dependence of the reflectance. Reconstructed image with an implemented model will have the spectral characteristics of the assumed model and may not be of a real material in-scene. The difference between the material mapper methods covered previously in this chapter and the pseudoinverse method is that in this method, the pixels in the scene are not substituted by the spectral library materials.

Matrix R and learning methodologies have proven themselves to estimate the reconstructed output well [90, 79]. By combining source separation with pseudoinverse method in Matrix R in (4.16), we get:

$$Y_{\text{train}} = MX_{\text{train}}$$

$$M \approx Y_{\text{train}} X_{\text{train}}^{\dagger} \tag{4.17a}$$

$$Y_{\text{test}} = MX_{\text{test}}$$

$$Y_{\text{test}} \approx (X_{\text{train}}^{\dagger} Y_{\text{train}}) X_{\text{test}}, \forall \{Y,X\}_{\text{train}} \supseteq \{Y,X\}_{\text{test}} \tag{4.17b}$$

Calibration data extracted from spectral library

Like the MR method the transformation matrix S can be obtained by matching pixels in the multispectral scene with that of the HSI spectra in the spectral libraries. Given a scene, the goal is to map the spectral content of individual pixels in the scene in terms of a linear combination of library materials where it is assumed to be comprehensively contains most, if not all, material species in the scene. In the pseudoinverse method shown in equation (4.17), the calibration data for the training scene, $(X^{\dagger}_{\text{train}} Y_{\text{train}})$, is extracted from the spectral library where 'X_{test}' is the MSI input for the test image to reconstruct, 'X_{train}' is the MSI image extracted from the library data and 'Y_{train}' is the HSI image of the material library.

Reconstruction accuracy of pseudoinverse method

Three experiments are designed to test for the accuracy of the pseudoinverse method. The first experiment extracts 50 materials from the USGS material library provided by JPL. The materials are subset to the centre wavelengths of HySpex VNIR-1600 sensor, whose spectral plot is shown in figure 4.7. The rest of this chapter uses band numbers instead for understanding of which bands are selected from the input and output. The relationship between the band number and wavelength is presented in figure 4.8 which shows that the spectral bands of HySpex VVNIR-1600 sensor is almost equally spaced. The correlation of one spectral band with respect to (w.r.t.) another is given in figure 4.9.

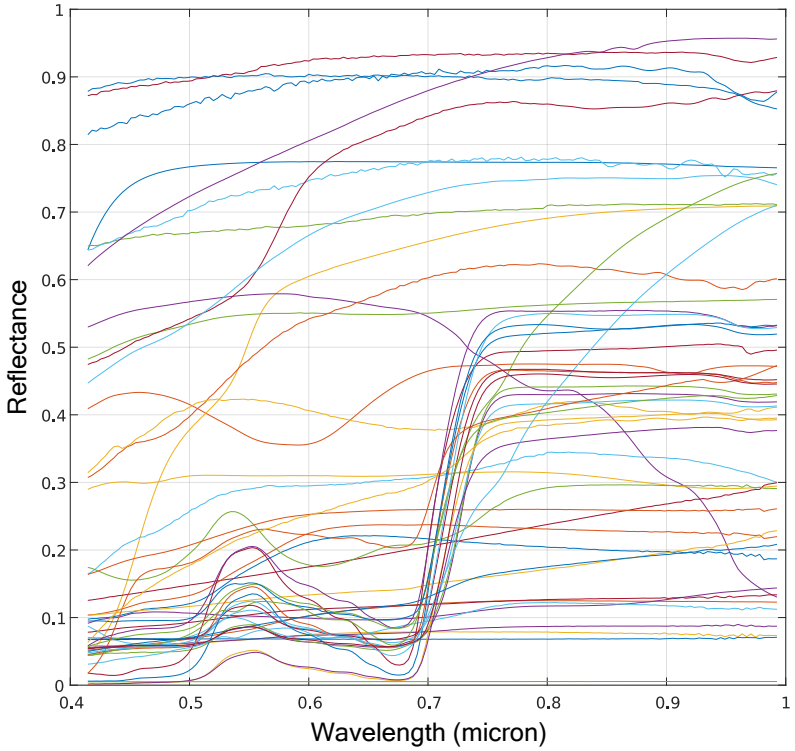


Figure 4.7: Spectral plot of the 50 library materials extracted from the USGS material library and subset to the centre wavelengths of HySpex VNIR-1600 sensor.

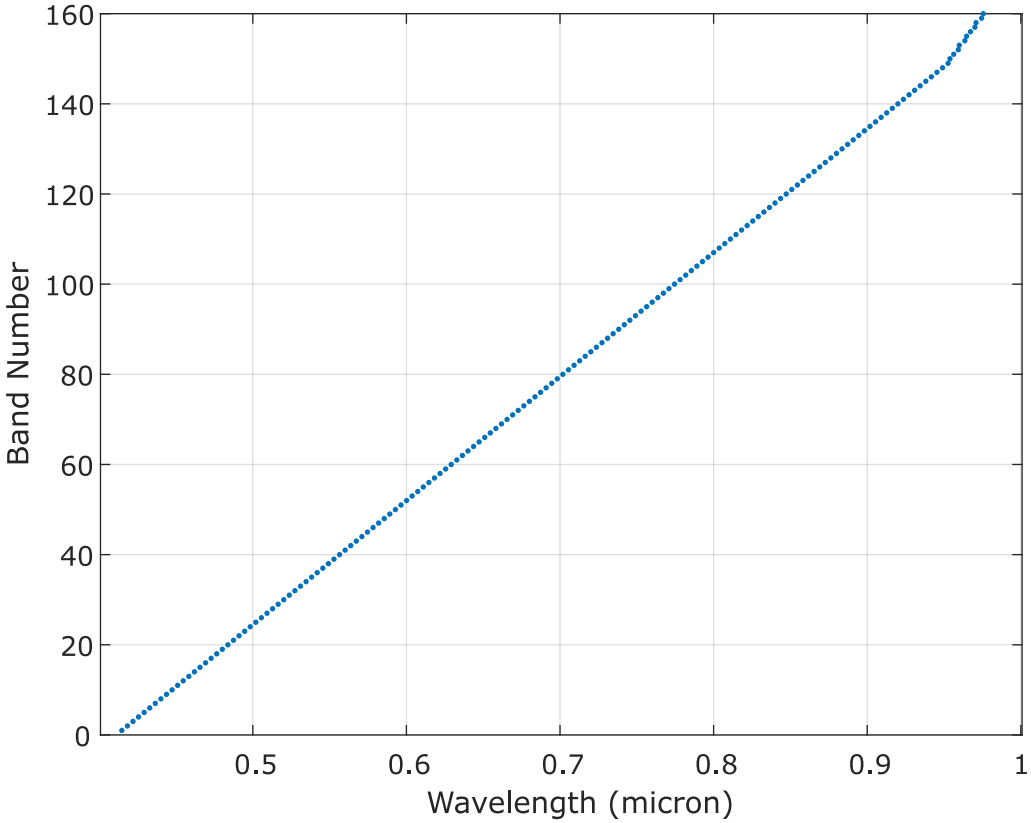


Figure 4.8: The relationship between band number and the wavelength, showing that the 160 bands of HySpex VNIR 1600 sensor is almost equally spaced.

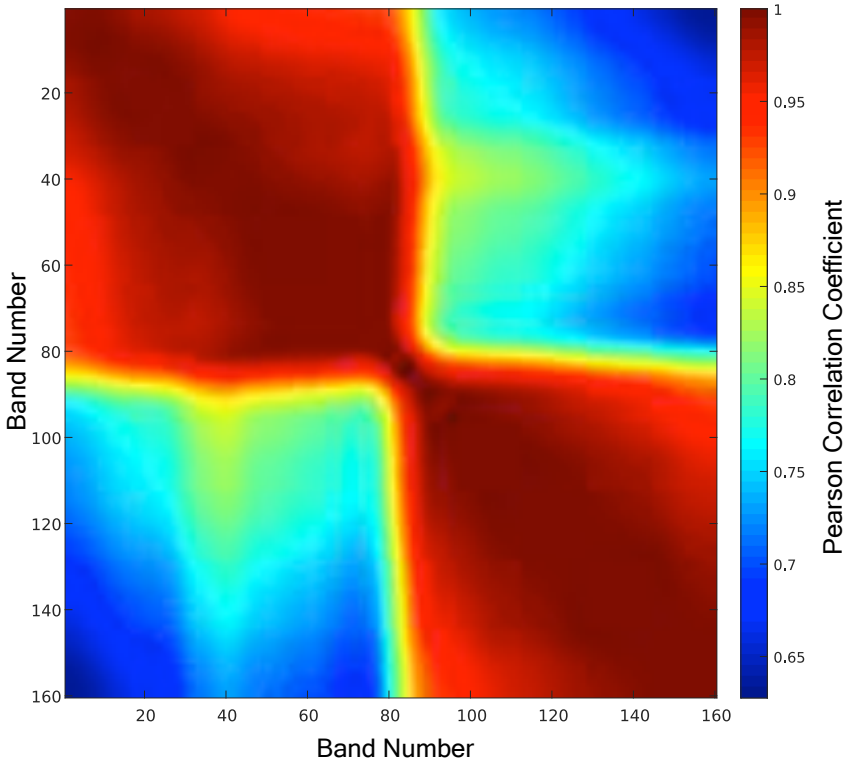


Figure 4.9: The Pearson correlation coefficient of one band with another with the 50 selected materials from figure 4.7.

With the correlation of different one band with another shown in figure 4.9, two measures are drawn for evaluation of the reconstruction. The first measure, showing one-to-one correlation or 'singleband correlation' is the maximum positive correlation that one band contributes to another. Singleband correlation of a given band 'k' w.r.t. a collection of bands in 'B' is given by:

$$C_s(B,k) = \max(PC(b_i,k)), \text{ where } b_i \in B \tag{4.18}$$

The second measure, showing many-to-one correlation or 'multiband correlation' measures the correlation of reconstruction where more than one band contribute with a weighting factor 'w'. The weighting factor can be calculated by the pseudoinverse method itself. The multiband correlation of a given band 'k' w.r.t. a collection of bands in 'B' is given by:

$$C_m(w,B,k) = PC(\sum_i w_i b_i, k), \text{ where } b_i \in B \tag{4.19}$$

The singleband and multiband correlation of the library dictionary with the 50 materials from figure 4.7 is shown in figure 4.10. A sample material (at index 30) is selected random from the collection of 50 materials for this experiment and was reconstructed with four spectral bands on input to the 160 bands of HySpex VNIR-1600 sensor. The location of the four input bands are changed from one run to another. It can be observed in figure 4.11 that the pseudoinverse method fails to reconstruct if the wavelengths are not spread properly i.e. output band is not highly correlated with at least 1 input band.

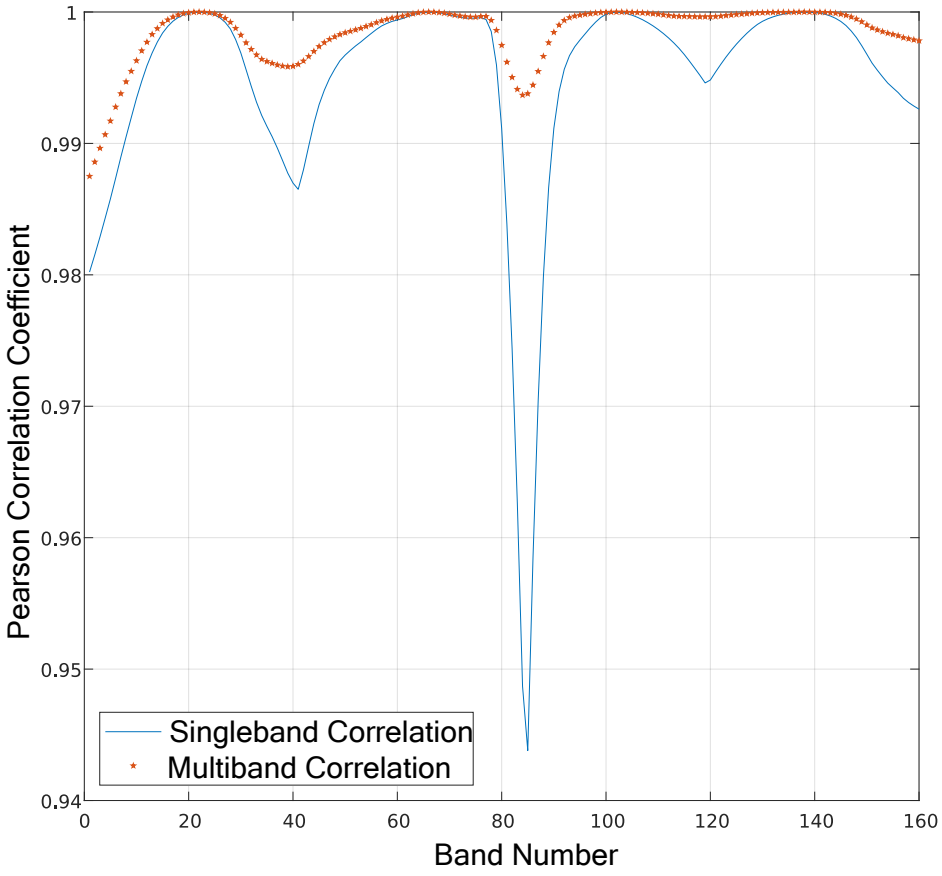


Figure 4.10: Correlation differences between singleband (one-to-one) and multiband (many-to-one) on the wavelengths of HySpex VNIR-1600 sensor. Four spectral bands are chosen at band numbers (22, 66, 102, 138) to represent multiband correlation in this figure.

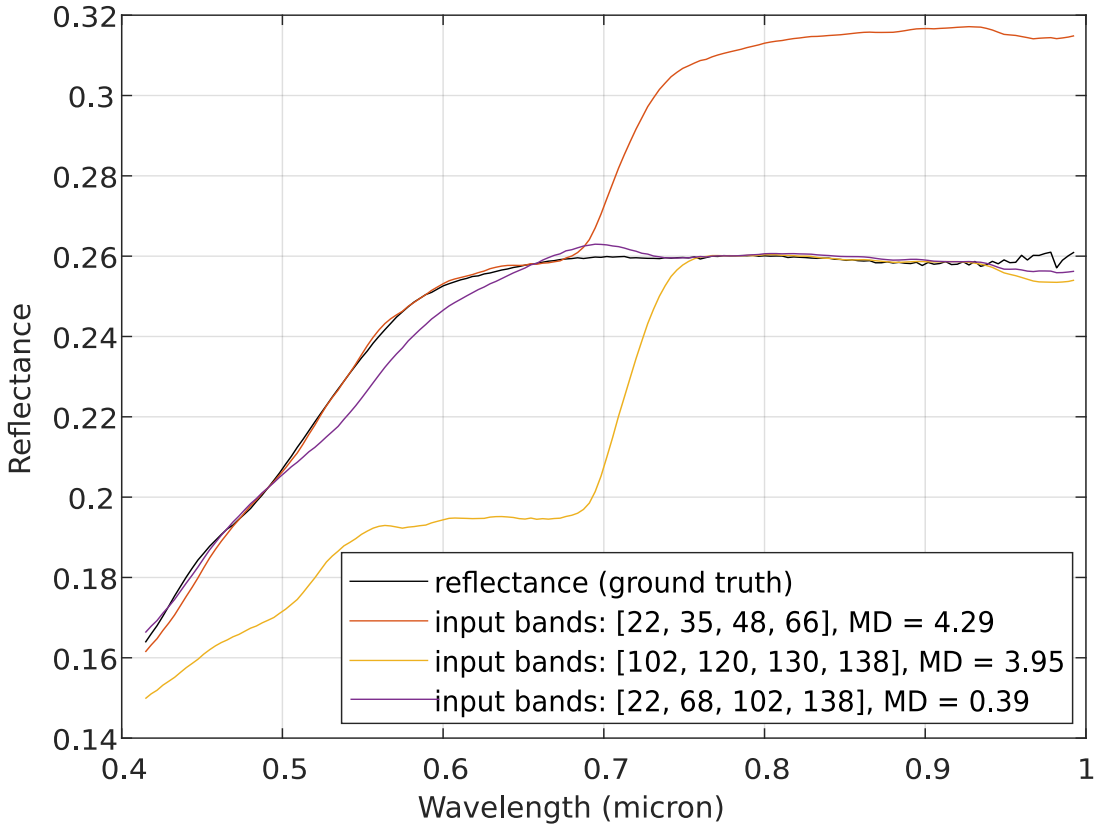


Figure 4.11: Spectral reconstruction using the pseudoinverse method with four input bands to the 160 bands of HySpex VNIR-1600 sensor. The location of the four input bands are changed from one run to another.

The second experiment to evaluate the reconstruction accuracy of the pseudoinverse method using the same 50 materials from the first experiment. However, for the second experiment, the sample that has low correlation with among the other 49 samples is selected for reconstruction. Figure 4.12 shows the Pearson correlation coefficient of one material with another. The material index 17 is selected as the test sample as it is relatively less correlated with the other 49 materials. The result presented in figure 4.13 shows the reconstruction fails in the absence of a material which has a low correlation with other materials in the training set.

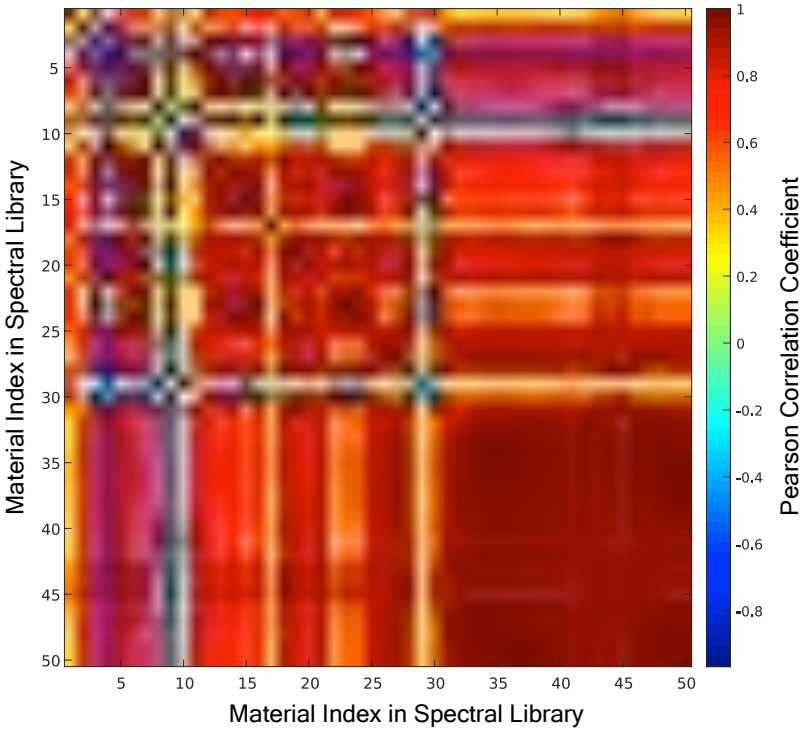


Figure 4.12: The Pearson correlation coefficient of one material with another with the 50 selected materials from figure 4.7.

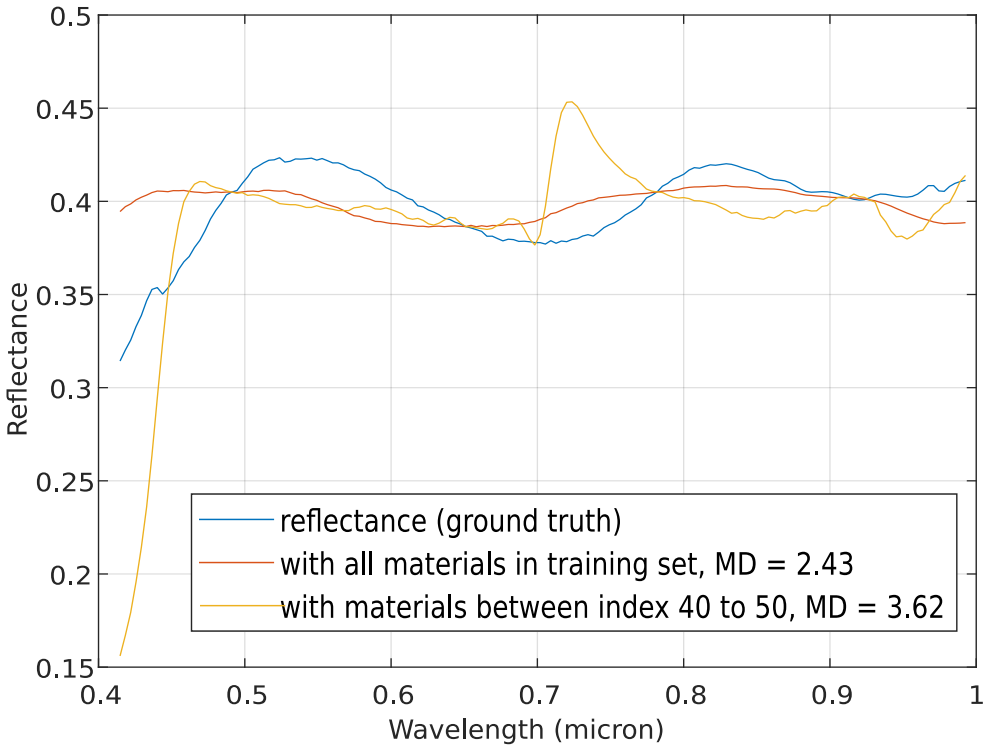


Figure 4.13: Reconstruction accuracy of low correlated material in library (figure 4.7, material index 17) with and without it being present in the training set. The spectral reconstruction is done from four band input (band numbers 22, 66, 102, and 138).

The third experiment extends the second experiment with all the materials in the collected library instead of selected 50 from first and second experiments. The combined library contains a total of 6821 materials with 70.33% vegetation, 19.23% minerals, and 3.87% artificial materials. The third experiment extends to all 448 bands of Selene H23 Dual scene, reconstructing from the centre wavelengths of WV3 satellite which has only 16 spectral bands. Spectral reconstruction is performed with the pseudoinverse method in four runs: the first reconstructing all 6.8k materials together, the second reconstructing only the vegetation materials with the vegetation materials in the training set, the third and fourth repeats the same conditions as the second run with minerals and artificial materials respectively. It is observed from figure 4.14 that vegetation materials are most correlated than other materials.

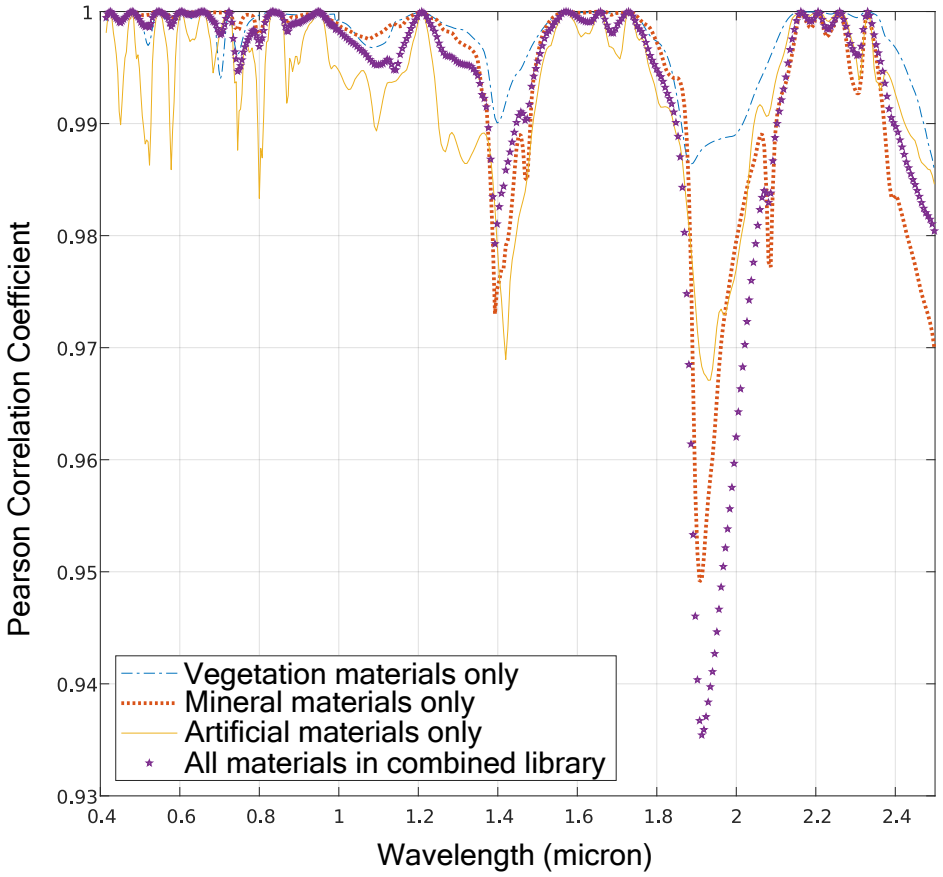


Figure 4.14: Observed pseudoinverse method reconstruction MD error against multiband correlation from centre wavelengths of WV3 sensor as input.

The mean Manhattan distance (sum of absolute point to point difference) is shown in figure 4.15, with the ℓ_1 norm (or Manhattan distance) of the mean are presented in

table 4.3.

Table 4.3: Reconstruction error of spectral library materials from WV3 to the wavelengths of Selene H23 Dual scene using pseudoinverse method.

Training and test set	MD error
All 6.8k materials	6.67e-04
Vegetation materials	2.71e-04
Mineral materials	3.91e-03
Artificial materials	2.29e-02

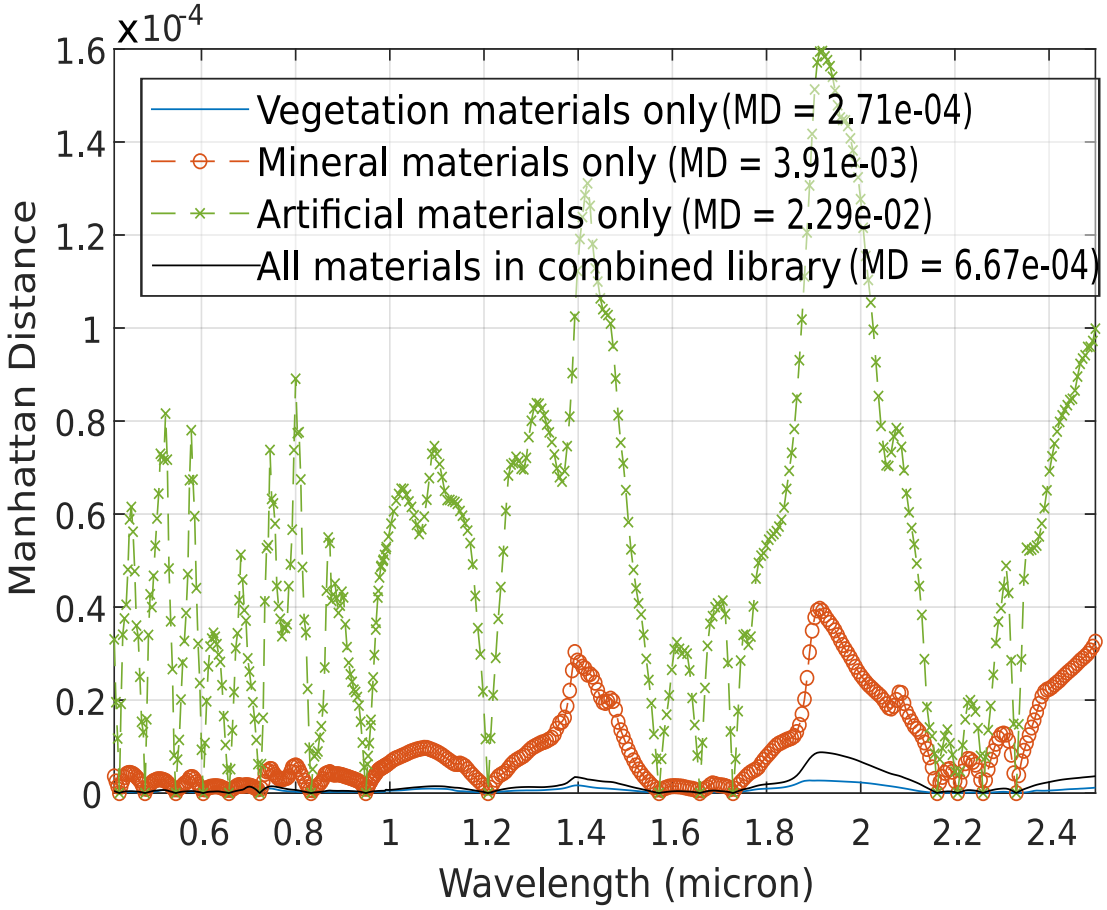


Figure 4.15: Spectral reconstruction error of library materials from the centre wavelengths of WV3 to the 448 bands of Selene H23 Dual scene using the pseudoinverse method.

All three experiments show that the pseudoinverse method demand high correlation between a given input and output. The correlation and reconstruction error relationship is summarised in figure 4.16.

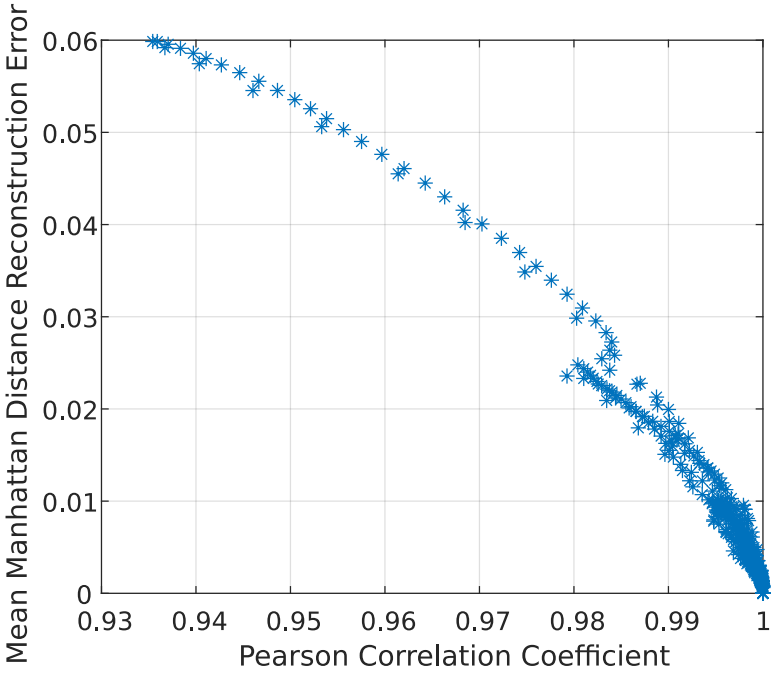


Figure 4.16: Average Manhattan distance reconstruction error (in reflectance data) against multiband correlation showing that decreasing correlation produces a larger error.

The reconstruction result in the Selene H23 dual scene shown in figure 4.17 show that vegetation materials, being the dominant material in Selene scene kept a relatively lower reconstruction error (within 8% for WV3 and 30% for WV2 input) compared to other materials in the scene, but still the method produces high reconstruction error. The reconstruction error in Selene is in line with the conclusion of the three experiments performed to evaluate the pseudoinverse method.

While pseudoinverse method for spectral reconstruction provides an approximation depending on the multi-band correlation and is proven to work in homogeneous collection of materials, the preservation of high dimensional statistic that HSI contain, is in theory, preserved by dictionary learning methods. Therefore, this thesis moves forward by proposing improvements in sparse coding dictionaries and adaptation of a dictionary learning method for material allocation.

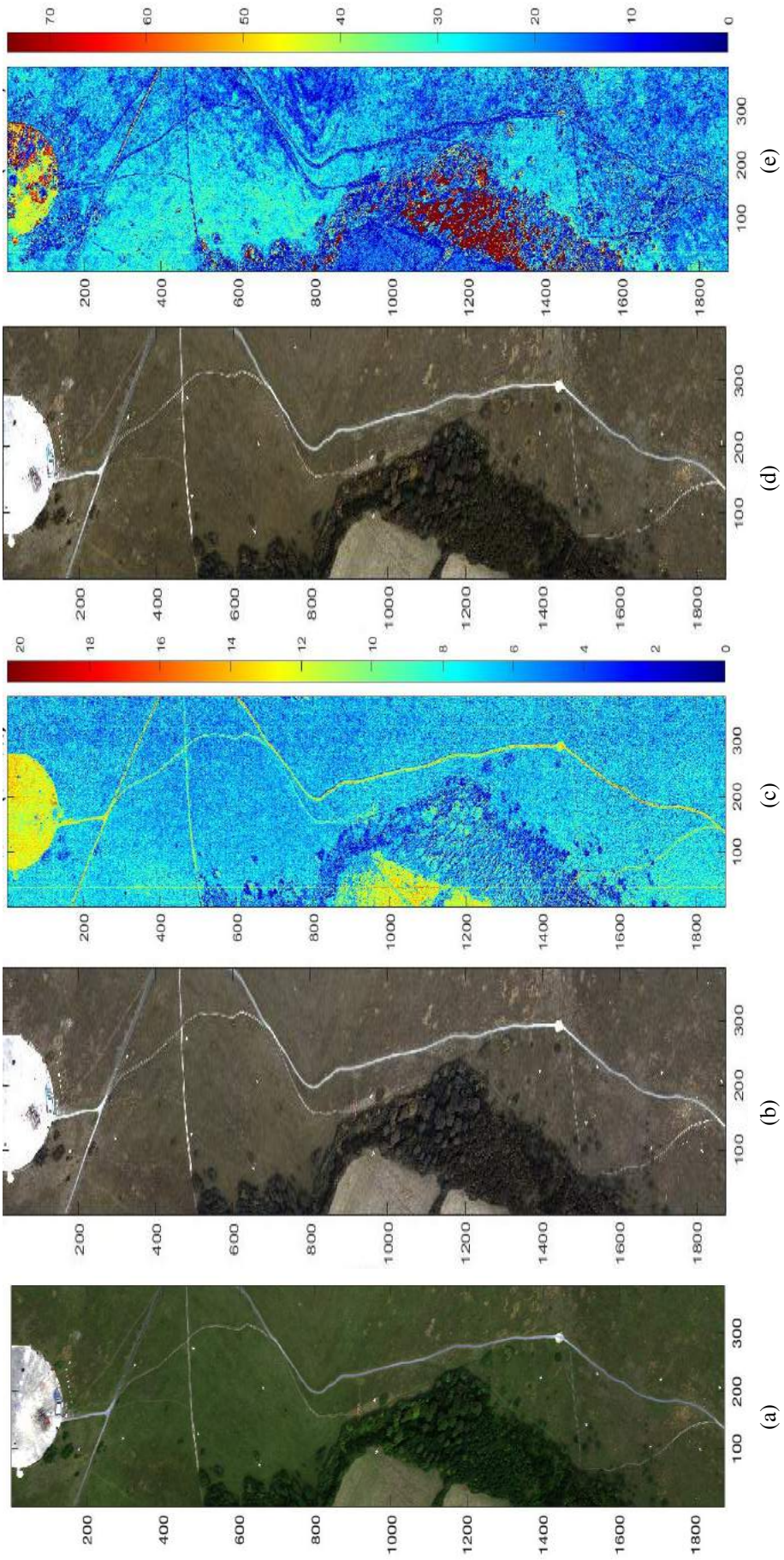


Figure 4.17: Pseudoinverse method for spectral reconstruction on Selene H23 Dual scene. The sub-figures (a) show the ground truth RGB, (b) and (c) the RGB and DL1NE error map of the reconstructed scene from WV3 input, and (d) and (e) are the RGB and DL1NE error map of the reconstructed scene from WV2 input. Figures (c) and (e) are with a mean of 6.78% and 24.79% respectively. The colorbar shown is upper bounded by 3x mean for visual purposes.

4.5 Spectral reconstruction from a learned dictionary

The optical reflectance of objects in the visible to short wave infra-red (SWIR) spectral region is sparse in the sense that their spectral characteristics are smooth and that they can be fully represented by a few basis function. To recall the dictionary learning and unmixing methods covered in chapter 2 and the subjects covered in this current chapter of this thesis, the special property of surface reflectance thus allows a range of compressive and filtering techniques, such as the PCA, Wiener estimation, sparse dictionary learning, Matrix R and inverse problem methods, to recover or to reconstruct the full narrow band hyperspectral characteristic of surfaces from imagery which contains only a few multispectral bands.

Sparse dictionary learning is a principled way that attempts to learn the basis function from a HSI imagery for training. This method works well when the statistics of the training data is close to that of the test data set and that the appropriate set of model parameters can be found faithfully, as evidenced from publications like [11]. This thesis, thus, concerns itself to proposing improvements to DL algorithms, and proposing an external material allocation procedure for a robust scene reconstruction in the next chapter.

4.6 Chapter summary

Multispectral images are relatively easier to obtain than hyperspectral data. Today, multispectral satellite images are collected more than once a day in some metropolitan cities. With a higher temporal resolution of multispectral satellite data, spectral reconstruction is an efficient way to obtain hyperspectral data (100s of spectral bands) from a given multispectral scene (usually $<$10 spectral bands).

Material allocations in Hyperspectral scene simulators like CameoSim takes multispectral image as input and to match them with a pre-compiled material library with a constraint of up to four material mixture in a pixel. Nearest matched materials and

texture material mapper (TMM) are popular choices among these simulators. Nearest matched material is limited to only one endmember per pixel and does not allow for material mixtures resulting in high reconstruction error.

Other spectral reconstruction include pseudoinverse method and the solution may be in local minimum. Unmixing using dictionary learning methods relies on (i) a hyperspectral training data which is a superset of endmember materials for the multispectral test data and (ii) the ability to reproduce the sparse representation matrix from multispectral input as accurately as with hyperspectral input. With the material limit in simulators, a constrained optimisation on the abundance matrix is required for the proposed material allocation method.

Chapter 5

Dictionary Learning Enhancements

This chapter focuses in two areas of work that enhances the robustness of scene simulations through (i) a refined dictionary learning technique and (ii) an algorithm which allows the generation of material species for scene simulation under severe constraints condition, such as the limitation of 4 material mixing per pixel. The first contribution addresses how the classic sparse coding dictionary (C-SCD) [11] that presented in chapter 2 can be enhanced such that a more robust dictionary can be learnt more faithfully. One main drawbacks in the C-SCD is the randomly selection of pixels from the scene for learning, which gives very variable performances even when the same scene is run repeatedly. The second contribution is the development of an algorithm which will allow CameoSim to perform better even when it only works under a maximum of 4 materials mixing per pixel. This chapter details the proposed enhancement to C-SCD algorithm, and a novel external material allocation module for CameoSim.

The objective is to propose a robust and efficient dictionary learning (DL) technique, which enhances HSI scene reconstruction with the given MSI or HSI data sets as inputs. This chapter proposes two different and blind approaches to overcome the drawbacks in random sampling, and adopts C-SCD to the material allocation structure as an external classification method for CameoSim. The proposed approaches are:

- sparse coding dictionary learning with simultaneous orthogonal matching pursuit (SCD-SOMP, [23]), and

- k-means sparse coding dictionary (KMSCD, [3]).

This chapter concludes with a proposed material allocation approach combining KMSCD with the fast non-negative orthogonal matching pursuit (FNNOMP) in section 5.3, and a proposed variant of OMP called rapid OMP in section 5.4.

5.1 Sparse Coding Dictionary with Simultaneous Orthogonal Matching Pursuit (SCD-SOMP)

To recall Chapter 2, the classic sparse coding dictionary (C-SCD) learning algorithm [11] uses the Bayesian form of the LASSO problem. The C-SCD assumes a Laplacian prior probability distribution on the representation 'a' due to the high kurtosis on the distribution of 'a' around zero. The C-SCD uses a variational method first optimising for the 'a' with a handful of randomly selected pixels from the scene, and then taking a gradient descent step on the dictionary 'D' using the least mean squares method.

In Selene, vegetation materials like grass and tree cover 95% of the scene and target materials like the Orange Perspex covers around 1% of the scene. Figure 5.1 shows the three locations of Orange Perspex materials present in the whole scene. And, learning of the trace materials is important detection applications. Anomaly detection algorithm like RX and a target detection algorithm like the ACE is dependant on the square of the Mahalanobis distance for its distance measure, which involves two in-scene parameters: (i) the inverse covariance matrix which is predominantly dependant on the dominating or background materials, and (ii) the spectrum of the individual target pixel. In random sampling of the pixels, and with the minimisation of the objective in $||Y - Da||_F^2$, the probability for a dictionary to learn a material is proportional to its abundance distribution in the training scene. To investigate the inconsistencies in the C-SCD to learn trace materials, Selene H23 VNIR scene was run for 50,000 (or 50k) iterations (figure 5.2 showing the 'D' at different iterations).

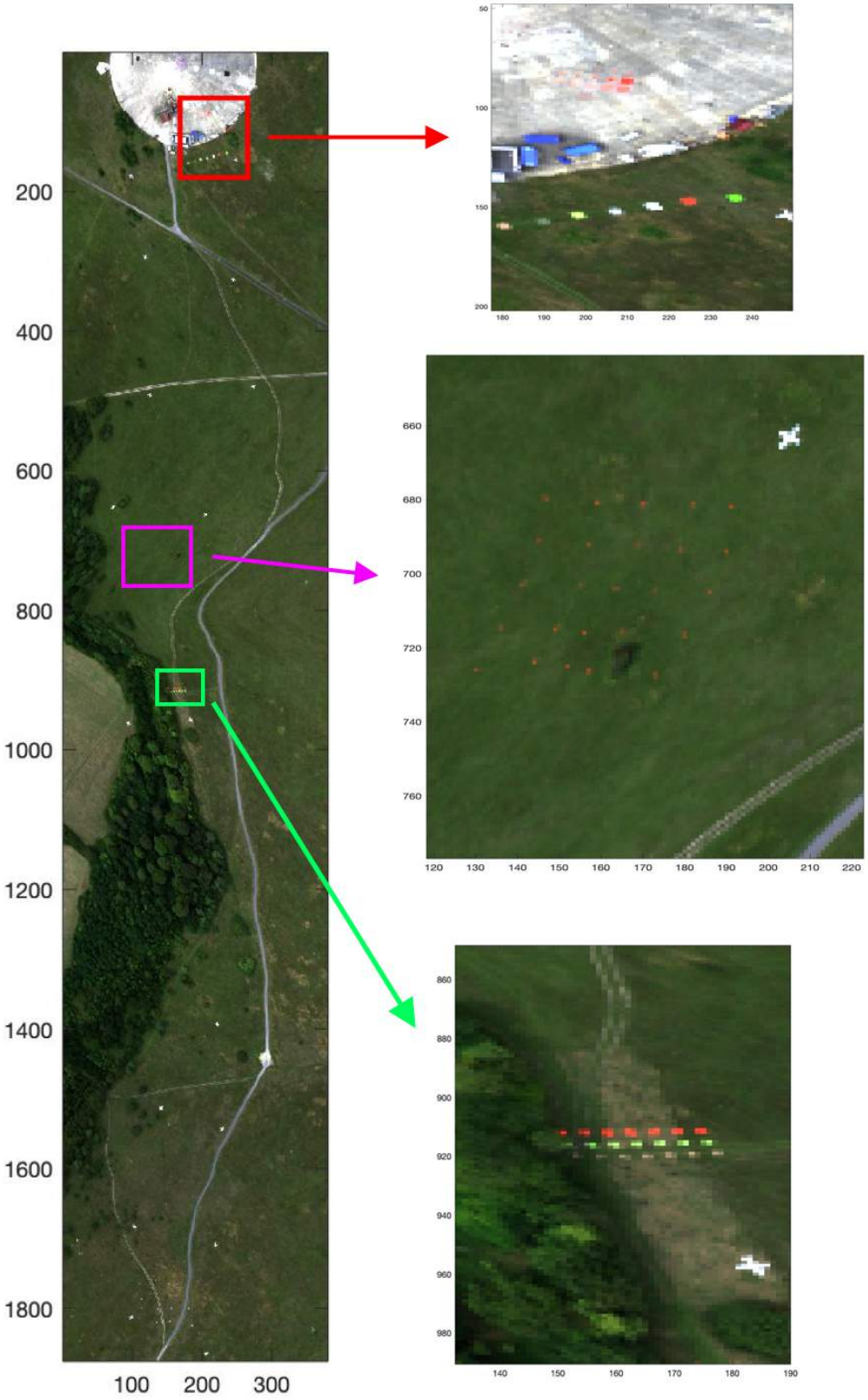
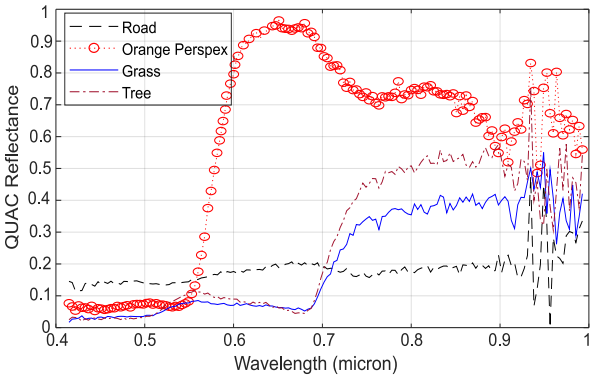
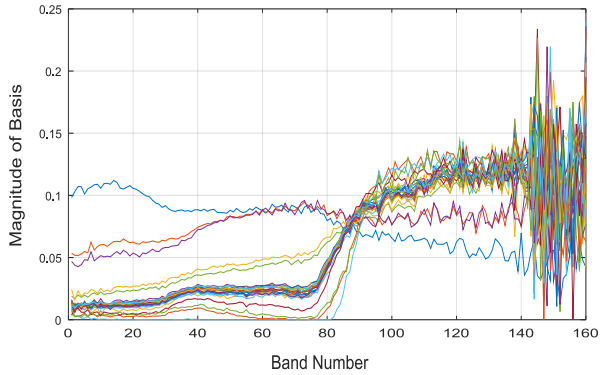


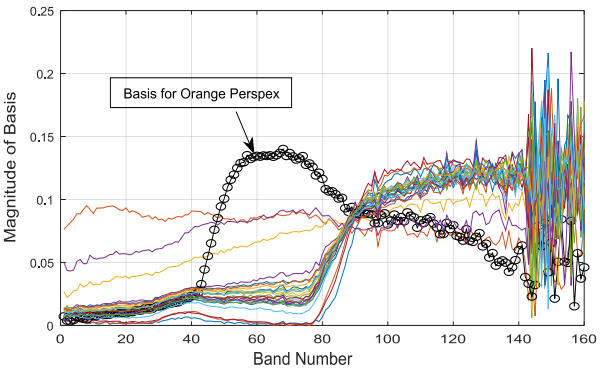
Figure 5.1: RGB image showing the three locations of Orange Perspex target material (the orange pixels) in Selene H23 Dual scene.



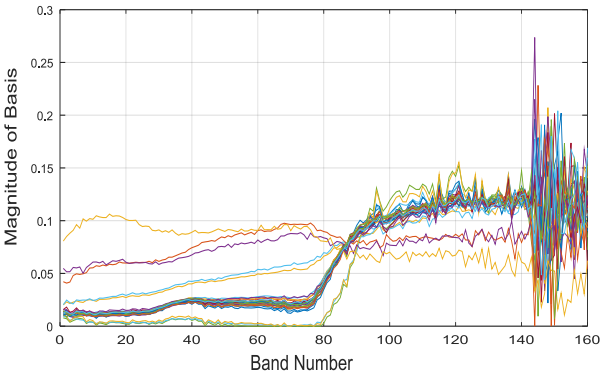
(a) Example spectra of different materials on ground



(b) Dictionary at 15k$^{\text{th}}$ Iteration



(c) Dictionary at 23k$^{\text{th}}$ Iteration



(d) Dictionary at 35k$^{\text{th}}$ Iteration

Figure 5.2: C-SCD algorithm's progression with iteration on Selene H23 VNIR showing that Orange Perspex material, once learned in 23kth iteration is lost in 35kth iteration. [23]

It is evident in figure 5.2 that the background materials are learnt well but trace materials once learnt were forgotten over the next few iterations due to their lower probability of existence in training samples in random sample selection. This results in the pixel with target material signature to be reconstructed by a mixture of background or high abundant materials in the training set. This results in reconstruction of target pixels to be a combination of materials which directly affects the accuracy of detection of the targets, as shown in the flow diagram in figure 5.3.

Algorithms that propose to address the issue of learning and detecting small targets with C-SCD model like in [91, 92] have proposed new frameworks to target detection for learned dictionaries and a precondition that the spectrum of the target materials are known prior to training. A priori information of materials is usually not known in real scenes before training a dictionary, and is not feasible when a large volume of HSI images require processing like the images in AVIRIS database.

The ultimate objective is to propose a dictionary learning (DL) technique that takes a step towards blindly capturing different elements in the scene, irrespective of the abundance distribution in the training set. The proposed technique aims to reconstruct pixels with trace materials with its own signature, and background materials with their spectral signatures. The proposed SCD-SOMP technique takes a step in that direction with the objective of minimising the infinite norm (or maximum value) of the residual error for a signal sample/pixel 'y' among 'Y' samples. With the minimisation of the infinite norm of a pixel in equation (5.1), one can at-least ensure that no pixel in the scene is left with a very high residual error compared to others. The objective of the SCD-SOMP for 'y' is thus:

$$\min ||y - Da||_{\infty}, \text{ where } y \in Y \tag{5.1}$$

To recall, the greedy algorithm presented in Chapter 2 for spectral unmixing, named SD-SOMP published in [14], constructs self-dictionary with in-scene pure pixel assumption by nominating the pixel with the maximum SOMP residue (estimated as

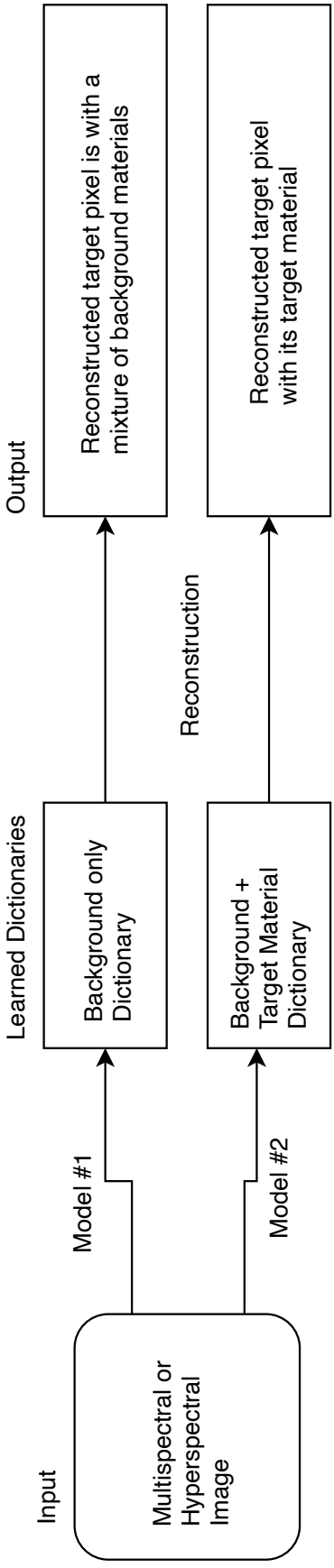


Figure 5.3: Reconstructed target pixel from two example dictionaries: model #1 that only focus on minimising the mean error with random sampling where the background materials are learned well and thus replacing target pixel with a mixture of background materials, and another (model #2) that incorporates learning of trace materials along with the background materials into its model reconstructing target pixel with its original target material.

$(I - DD^{-})y$ for dictionary 'D' and sample 'y') to be an end-member for the given iteration, where SOMP residue estimates the residual error of all dictionary atoms simultaneously. The algorithm continues till the pixel selected in 'k*th*' iteration satisfies $||y_k - D_k a||_2 \leq$ acceptable error (the dictionary 'D_k' are all the pixels selected previously in $(k-1)$ iterations and representation 'a' is constrained to sum-to-one). The algorithm goes on to automate the stopping tolerance term with error estimates from HySime algorithm [40].

In a real scene due to non-linear mixture, scattering of materials, and colour noise, the existence of an ideal pure pixel of all EM materials in scene is non-existence. With SOMP residue estimating the fitting error of all dictionary atoms simultaneously, the proposed sampling technique uses SOMP residue and selects the pixels with the highest magnitude in residual error at a given training iteration. The pseudo-code of the proposed SCD-SOMP is written in algorithm 10.

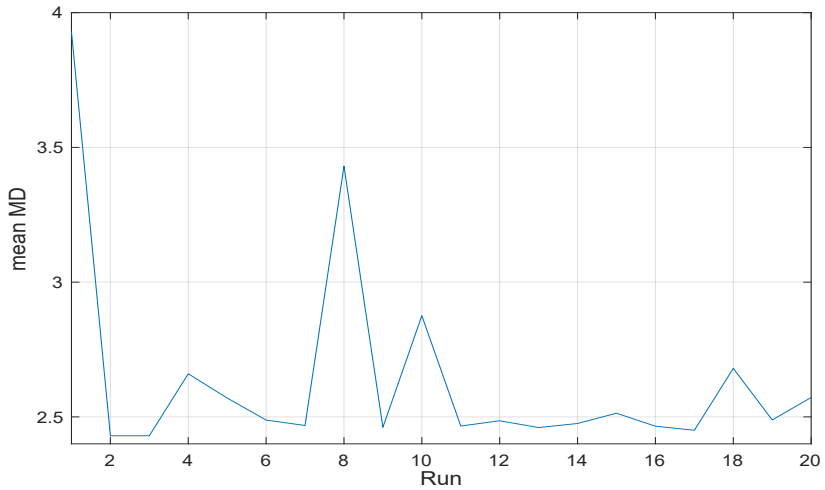
Algorithm 10 Proposed SCD-SOMP algorithm

1: **Import** HSI Image as 'Y', number of dictionary atoms 'n', user-defined step size 's' and step-size decay 'd', and maximum number of iterations as 'M'
2: **Initialize** dictionary as 'D' as 'n' random numbers
3: **for** `k = 1 to M` **do** ▷ Iterate till convergence
4: r = $||(I - DD^{-1})Y||_p$ ▷ Estimate error residue (SOMP residue)
5: Sort and select Y_k samples with starting with the maximum 'r'
6: $a_k = \min ||Y_k - Da_k||_p$ ▷ Infer coefficients
7: $D_k = (Y_k - Da_k)a_k^T$
8: $D \leftarrow D + sD_k$ ▷ Dictionary Update
9: s = s d ▷ Step size update

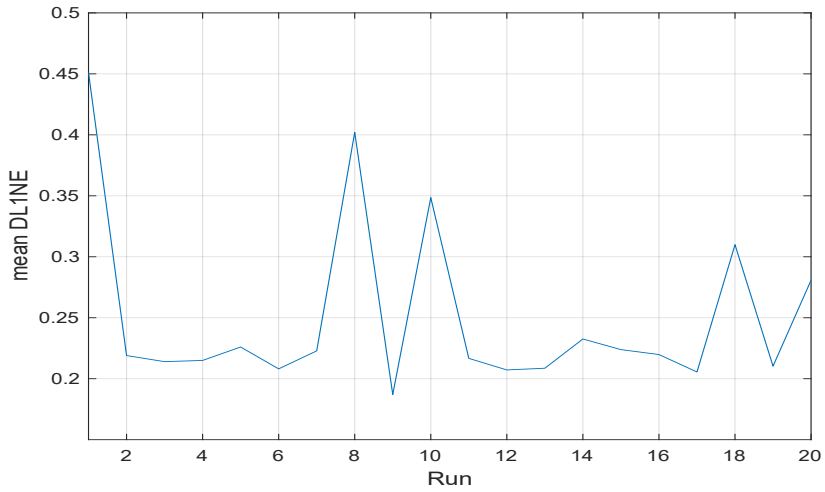
5.1.1 Experimental setup

This section describes the results of the C-SCD algorithm compared to the proposed SCD-SOMP. Two experiments are performed to test the reconstruction accuracy of the proposed SCD-SOMP compared to the C-SCD. The first experiment tests the reconstruction result with the increasing order of the number of atoms from a small count of 10 atoms to 50 atoms. The second experiment performed, tests for the

reproducibility/repeatability of the results running both algorithms for five repeated runs under the same training conditions with 40 atoms in dictionary. With a ≈1% mean error difference between five and ten runs (figure 5.4), five runs is sufficient demonstration of the repeatability errors.



(a) Manhattan distance (MD)



(b) differential L1 norm error (DL1NE)

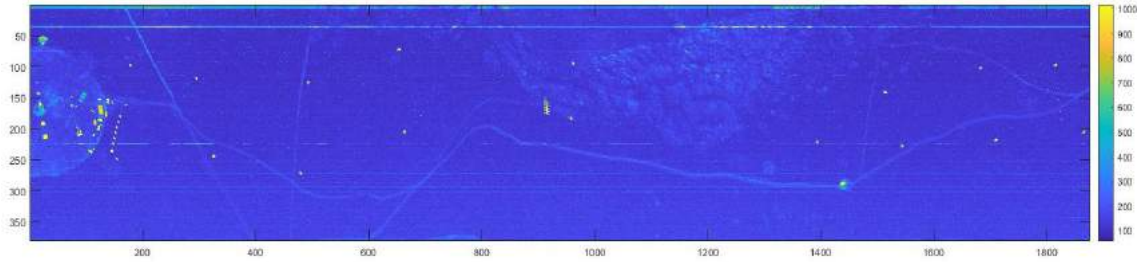
	MD		DL1NE	
Runs	mean	std	mean	std
1 to 5	2.8e+0	6.37e-01	2.65e-01	1.04e-01
1 to 10	2.77e+0	5.10e-01	2.69e-01	9.43e-02
1 to 20	2.64e+0	3.80e-01	2.50e-01	7.20e-02

(c) tabular summary showing the numerical accuracy of repeat runs

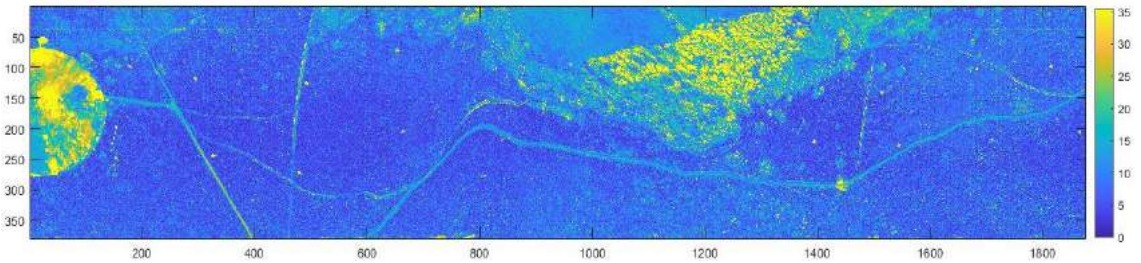
Figure 5.4: Error differences for twenty repeat runs with 40 dictionary atoms under the same training conditions. Results show a difference of ≈1% with ten runs compared to five runs, and ≈5.8% with twenty runs.

5.1.2 C-SCD vs SCD-SOMP: Reconstruction of trace materials

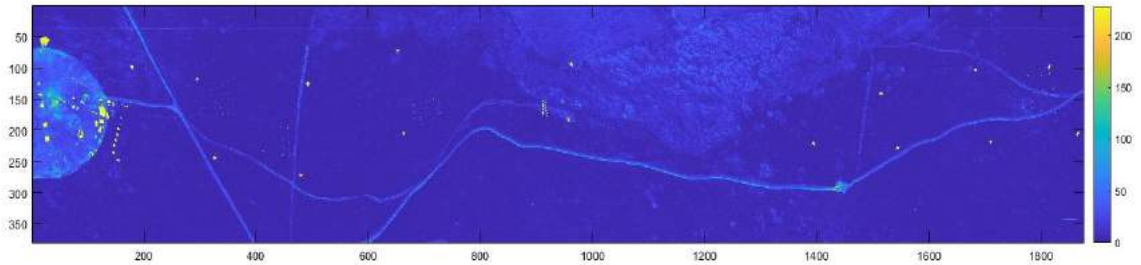
This section describes the ability of the DL algorithms, precisely the C-SCD and the SCD-SOMP algorithm, for reconstruction of trace materials in the scene. There are small amount of full pixel (i.e. material abundance = 1) artificial materials embedded in the scene and Orange Perspex target material covers $<1\%$ of the whole scene. Due to the random selection of the pixels in the C-SCD algorithm, the recovery of trace materials in the scene seem to fail as seen with the Reed-Xiaoli (RX) anomaly detection in figure 5.5.



(a) RX anomaly detection on the Ground Truth (GT) image



(b) RX on the reconstructed image of C-SCD with a correlation of 0.28 with GT



(c) RX with the proposed SCD-SOMP showing a Pearson correlation of 0.89 with RX GT

Figure 5.5: False colormap depicting RX anomaly detection with 10 dictionary atoms of the C-SCD and SCD-SOMP algorithms compared with the ground truth (GT). The colormaps are scaled to [0 to mean+3x standard deviation] for visual presentation.

To quantify the RX anomaly detection results, Pearson correlation coefficient is measured between the Ground Truth (GT) and the reconstructed results. It is seen quite clearly from figure 5.5 that the proposed SCD-SOMP delivers a better detection results,

with a 0.89 in correlation compared to C-SCD with a correlation of 0.28. The distance measure in the RX algorithm is the square of the Mahalonobis distance, and correlation of RX is used as a measure to match with the Ground Truth (GT) RX. A high correlation of the reconstructed data with GT, shows that RX of the background reconstructed has produced closer to zero and trace materials with higher values. The detection of trace materials is better quantified by the Receiver Operating Characteristics (ROC) curve. The ROC with ACE for detection with Orange Perspex material for the same reconstructed scene is presented in figure 5.6, which shows a 58% increase in detection accuracy (measured by area under the curve (AUC)) with the proposed SCD-SOMP compared to the C-SCD.

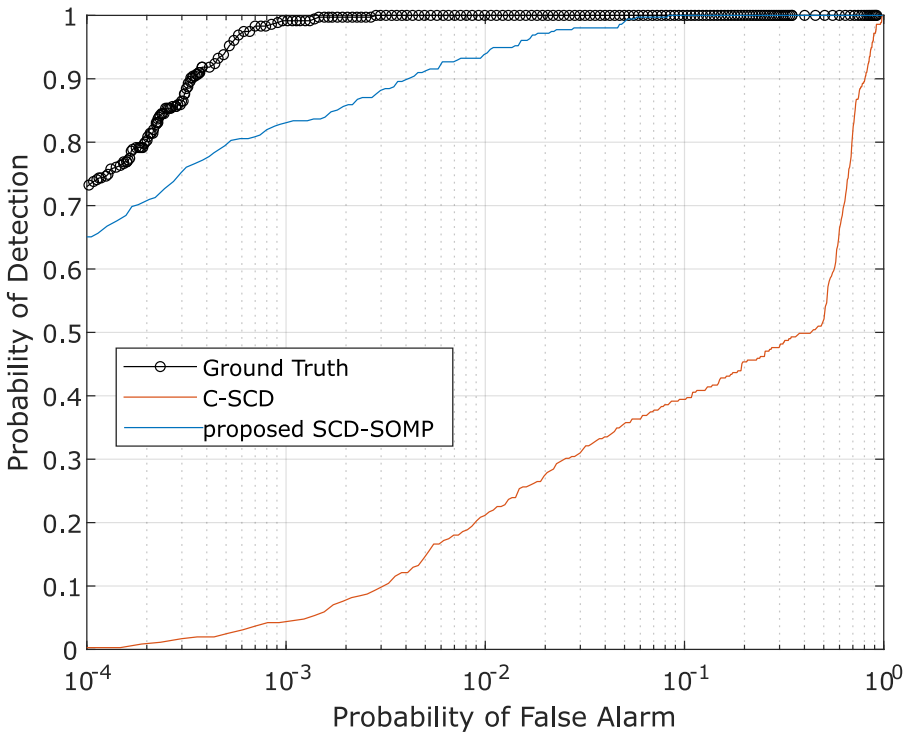
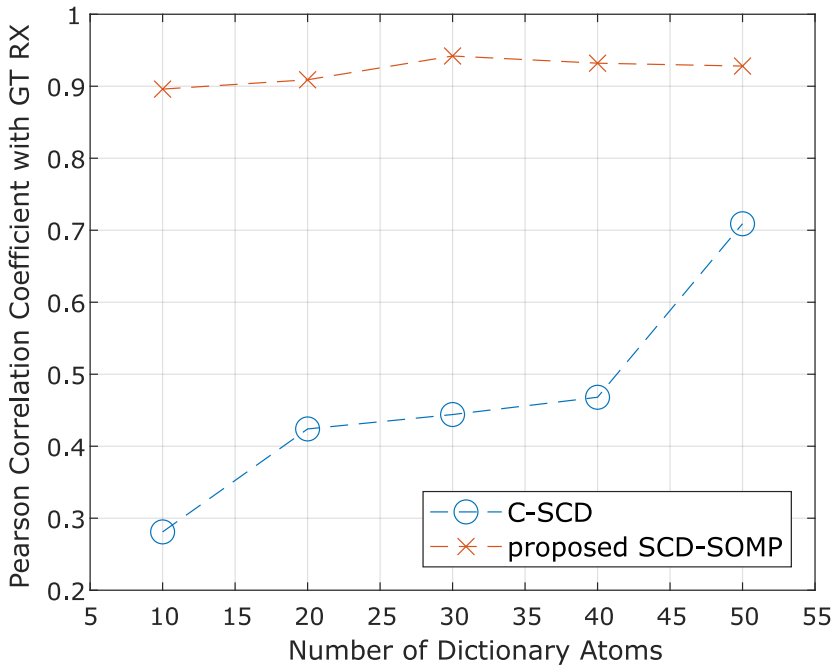


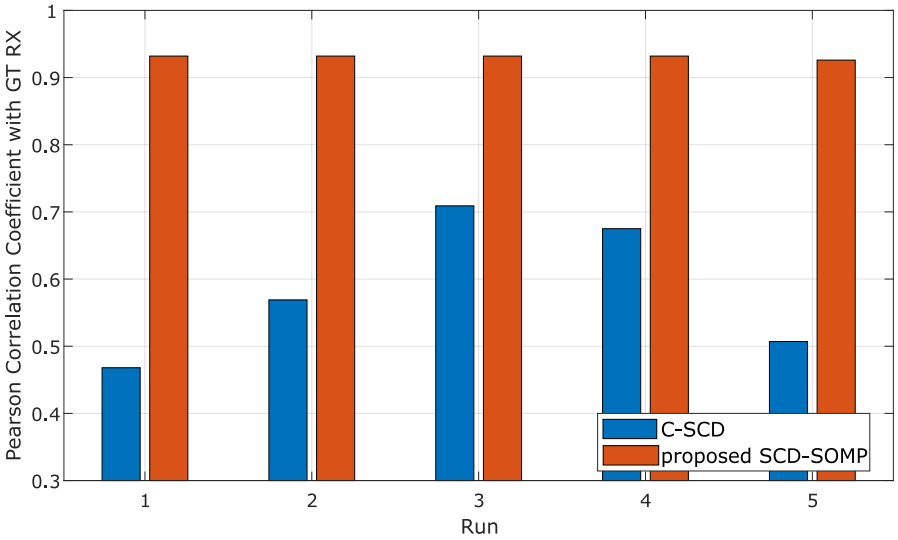
Figure 5.6: Receiver Operating Characteristics (ROC) curve using ACE for detection with Orange Perspex as target material for the scene whose RX anomaly detection is shown in figure 5.5. Results show a 58% increase in the area under curve (AUC) of the proposed SCD-SOMP (AUC = 0.98) compared with the C-SCD algorithm (AUC = 0.63).

The first experiment with increasing number of dictionary atoms show that C-SCD improves from a correlation of 0.28 with 10 atoms to 0.71 with 50 atoms, and the proposed method with 0.89 with 10 atoms to 0.93 with 50 atoms. The result presents an overall ≈100% mean improvement over C-SCD with ≈8.3x improvement in the

standard deviation. And, over the five runs SCD-SOMP has a 43% mean improvement over C-SCD with ≈73x improvement in standard deviation. The results are presented in figure 5.7 with detailed numerical accuracy in appendix A.1.



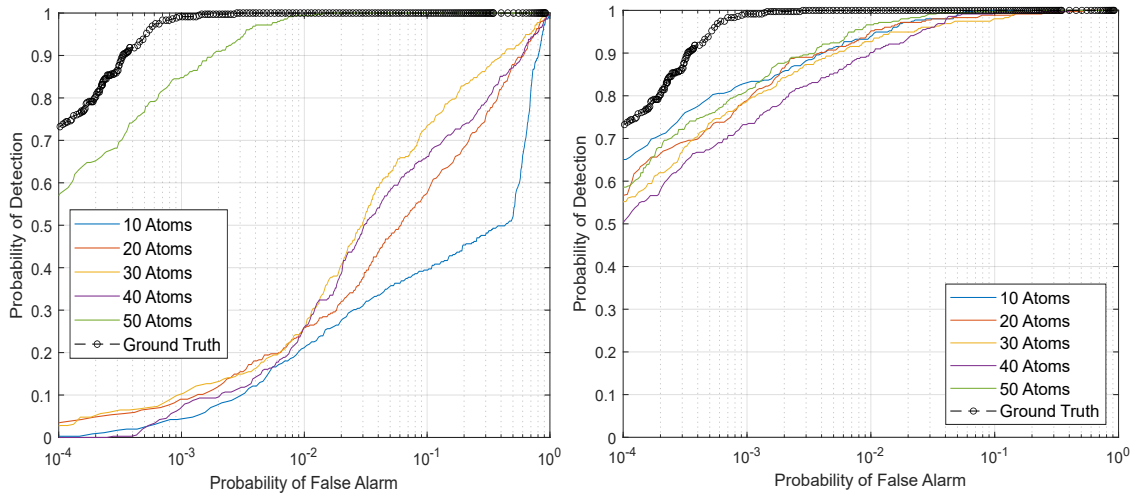
(a) RX detection accuracy with increasing number of dictionary atoms. SCD-SOMP show a ≈100% mean improvement and ≈8.3x in standard deviation over C-SCD.



(b) Five repeated runs with 40 dictionary atoms. SCD-SOMP show a ≈43% mean and ≈73x improvement in standard deviation over C-SCD.

Figure 5.7: Pearson correlation coefficient between the ground truth image and the reconstructed image performed by C-SCD and the proposed SCD-SOMP algorithms. The figure (a) shows the accuracy with increasing number of dictionary atoms, and (b) shows five runs under the same training conditions.

Similar improvement in the detection is seen in target detection with Orange Perspex as target material using the ACE algorithm. The first experiment, which is with increasing order of dictionary atoms shown in figure 5.8, show a ≈20% improvement in the mean area under curve (AUC) with a ≈70x improvement in the standard deviation over C-SCD. The second experiment, which is five repeated runs under the same training conditions with 40 atoms shown in figure 5.10, show a ≈7% mean improvement in AUC with a ≈92x improvement in the standard deviation. The detailed numerical accuracy is presented in appendix A.2.



(a) C-SCD (b) proposed SCD-SOMP

Figure 5.8: Receiver operating characteristics (ROC) curve of ACE detection for Orange Perspex material with increase in the number of atoms. The SCD-SOMP show a ≈20% improvement in the mean and ≈70x improvement in the standard deviation over C-SCD.

The jump seen in the ROC curve figure 5.8a for 40 to 50 atoms in C-SCD may present further evidence of the randomness nature in the existing algorithm. This can be seen more clearly from the figure 5.9 which shows an AUC of 9.99e-01 in the first run, however, the subsequent repeated runs under the exact same configuration fails to reproduce the result of the first run.

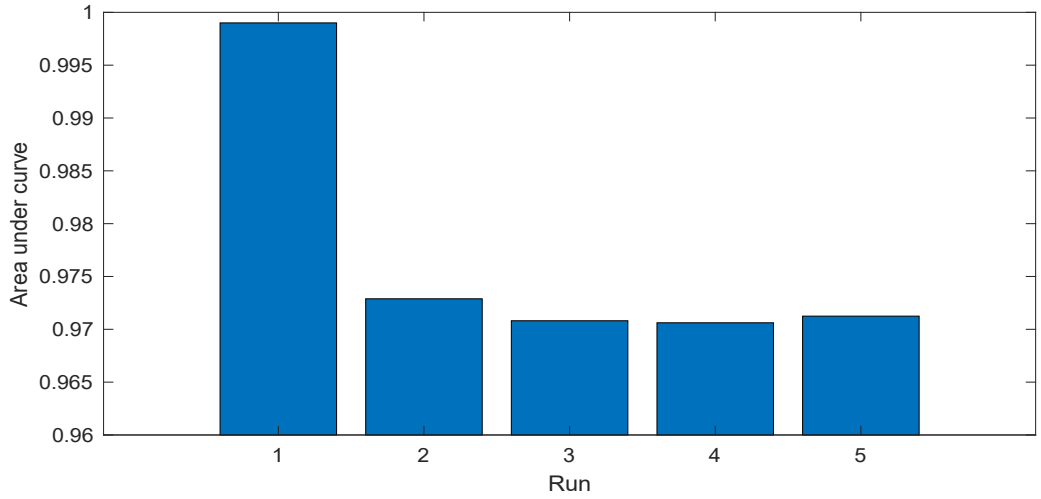
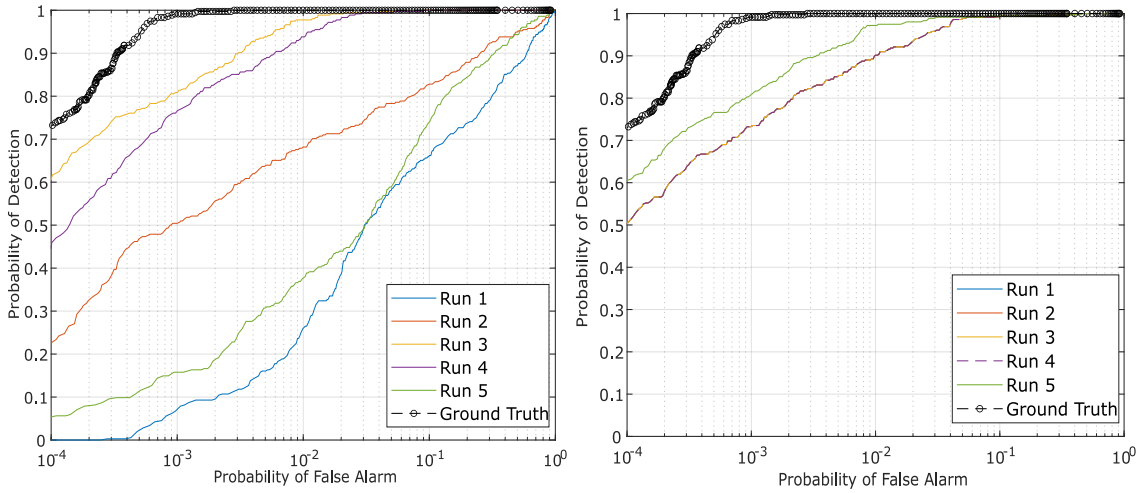


Figure 5.9: Area under curve (AUC) of five repeated runs with 50 dictionary atoms. Results show a mean AUC of 9.76e-01 with a standard deviation of 1.24e-02.



(a) C-SCD (b) proposed SCD-SOMP

Figure 5.10: ROC curve of five repeated runs with 40 dictionary atoms. Results show that the SCD-SOMP has a ≈7% mean improvement in AUC with a ≈92x improvement in the standard deviation over C-SCD.

5.1.3 C-SCD vs SCD-SOMP: Reconstruction of the background pixels in the scene

The improvements in the consistency and reproducibility of SCD-SOMP against C-SCD, and improvements in anomaly and target detection is seen in the previous section of the results. However, minimising for infinite norm in order to learn different materials

leaves room for improvement in the overall mean error. The first experiment with increase in the number of dictionary atoms (displayed in figure 5.11) show that C-SCD has a mean error of 2.91e-01% DL1NE compared to SCD-SOMP with 3.28e-01%. The C-SCD has shown it has ≈11% improved result than SCD-SOMP.

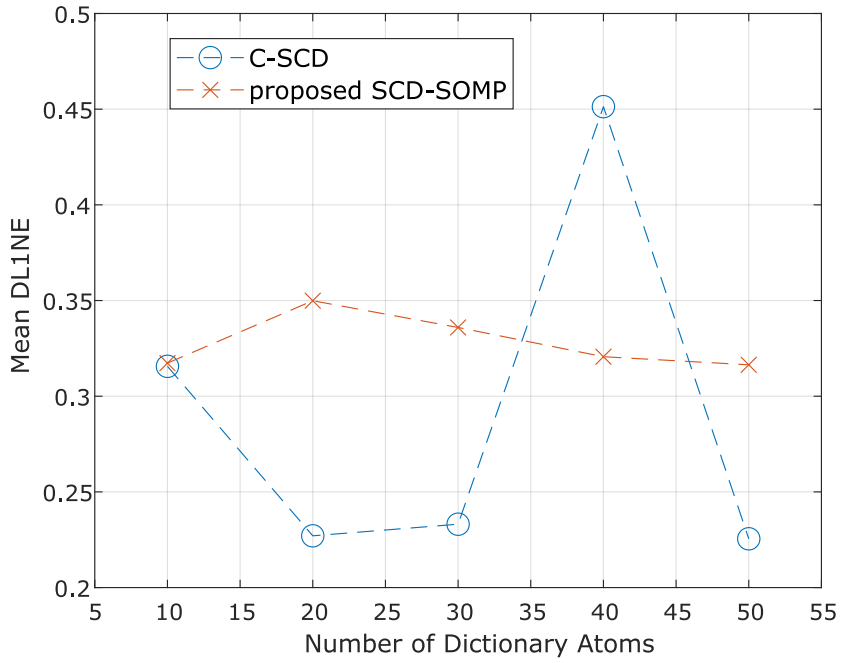
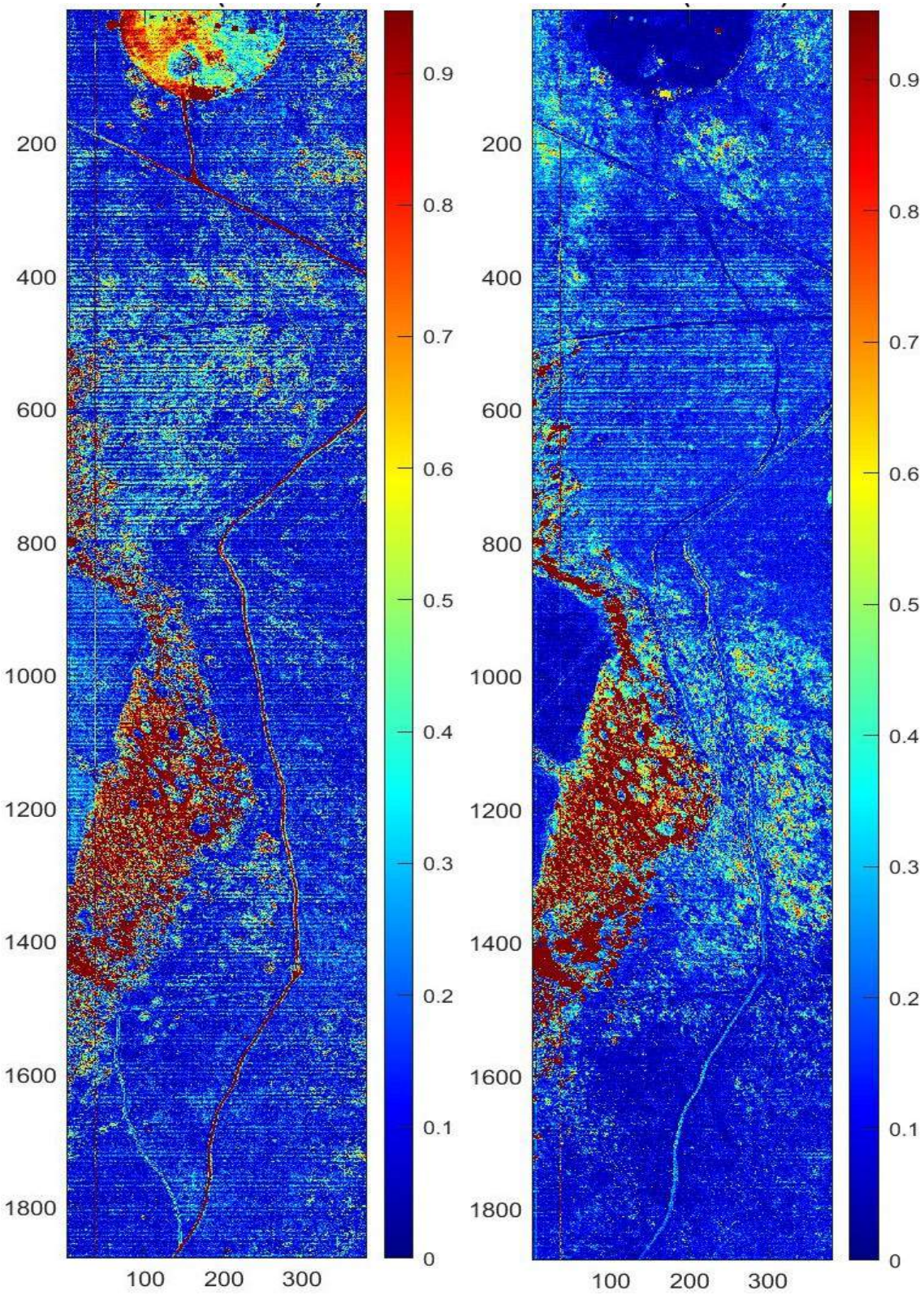


Figure 5.11: Mean DL1NE error against increasing number of dictionary atoms. The result shows that although the proposed SCD-SOMP algorithm has produced improvements in detection accuracy, the mean error is almost constant. The figure also shows the variation in the C-SCD algorithm.

An example of the DL1NE error map is presented in 5.12, which shows that C-SCD has produced ≈0.6% lower error than the proposed SCD-SOMP. To overcome the minimum mean error estimates, unmixing and material allocation is proposed with KMSCD, which is presented in section 5.2.



(a) C-SCD

(b) proposed SCD-SOMP

Figure 5.12: False-colour map with 10 dictionary atoms on the reflectance data (for the same result whose RX anomaly detection map is presented in figure 5.5) of DL1NE using C-SCD (left) with a mean error of 3.15e-01% and the proposed SCD-SOMP (right) with a mean error of 3.17e-01%. The colormaps presented range from [0 3×mean(DL1NE)].

5.2 K-Means Sparse Coding Dictionary (KMSCD)

The proposed k-means sparse coding dictionary (KMSCD) algorithm is presented in this section. The original C-SCD algorithm reveals that the technique for selecting atoms during the learning process is not well defined, resulting in the re-selection of atoms already selected or rejected in later iterations, resulting in a significant slowdown of the convergence algorithm. [48]. Although the C-SCD achieves a reasonable result, the complexity of the C-SCD learning is large $\approx$O(NK^2), where 'N' is the number of pixels in the scene, and 'K' is number of training iteration.

Two algorithms are proposed here, and both adopted a clustering algorithm (K-Means) to obtain a more representative spectral structure of the scene. The proposed technique reduces the computational complexity of the DL algorithm and has the ability to achieve better spectral reconstruction not only for the background pixels (i.e., most materials in the scene), but also the recovery of minority pixels (i.e. trace materials in the scene). The number of clusters (J_c) have been designed to be a few times greater than the intrinsic dimension (i.e., the number of EMs in the scene) to include the small amounts minority pixels in the scene to be clustered in its own class, and the decrease in the mean error with increase in the number of classes is presented in figure 5.13.

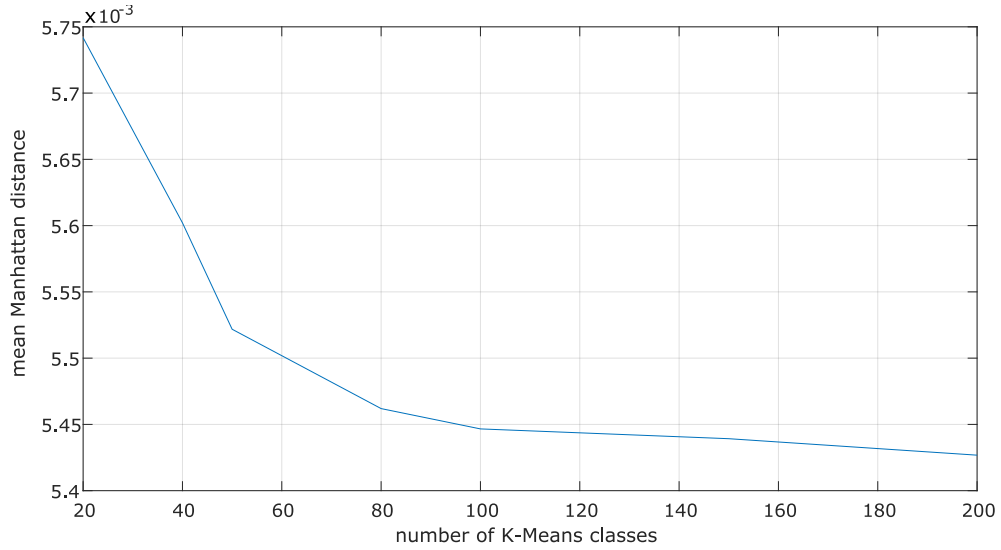


Figure 5.13: KMSCD with 40 dictionary atoms showing the decrease in the mean Manhattan distance error with increase in the number of KMeans classes.

These cluster centres set Ci , and are then formulated as the scene's initial dictionary for learning the algorithm. This is in great contrast to the conventional DL algorithms, which use all pixels in the scene to create to construct their comprehensive dictionaries. Typically, the number of pixels (N) in the HSI scene is in the order of hundreds of thousands, compared to the minimal number of elements (Jc) in Ci which is in the order of about one hundred pixels, the learning loop using the Ci set would automatically reduce the computational complexity by several orders of magnitudes.

KMeans algorithm iterates between cluster centres and membership. With a complexity of O(Nki) where 'N' is the total number of pixels, 'k' is the number of KMeans classes and 'i' the iteration number. Unless the size of the mini-batch samples required for training is the same as the number of atoms in the dictionary, selecting KMeans to initialise the dictionary requires the algorithm to be run twice: once for the initial dictionary and another for sampling adding a total complexity of O(Nk_1i) + O(Nk_2i). The result shown in figure 5.14 shows that the added complexity produces an increment of $\approx$0.68% compared to random initial dictionary with numerical accuracy in table 5.1.

Table 5.1: KMSCD with different initial dictionary and mini-batch sample size.

Initialisation	**no. of atoms**	**mini-batch size**	**KMeans run**	**mean MD error**
centroid	40	40	once	5.56e-03
centroid	40	200	twice	5.39e-03
random	40	200	once	5.42e-03

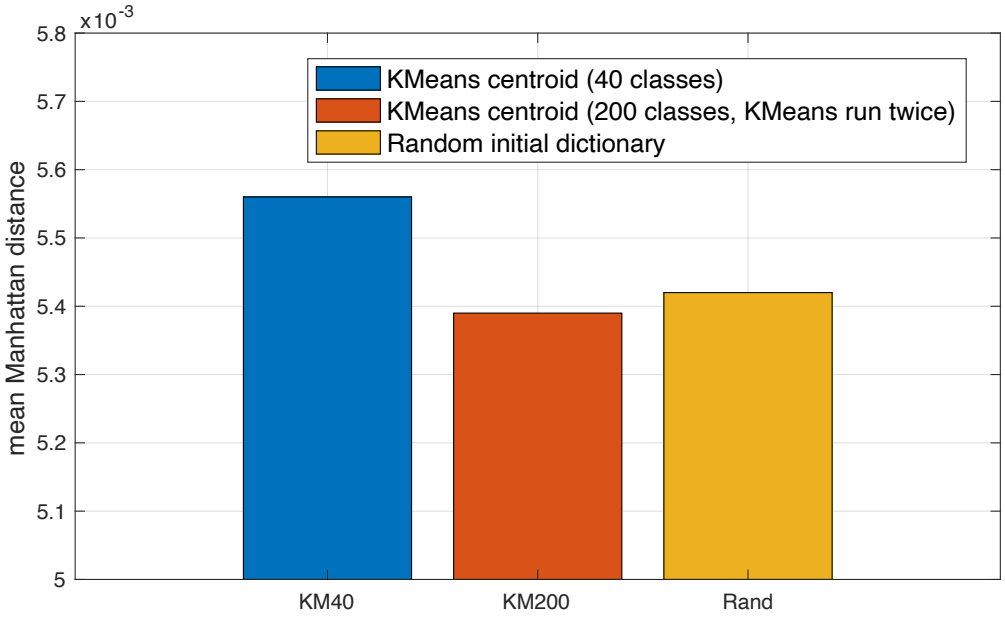


Figure 5.14: Mean Manhattan distance error differences between three initial dictionaries for KMSCD algorithm: KMeans algorithm run once where the number of training samples is the same as number of atoms and the initial dictionary is the centroid (KM40), KMeans algorithm run twice where the second run is to obtain centroid of a larger sample size (KM200), and random initial dictionary (Rand). Rand shows a 2.5% improvement in the mean error over KM40 where KM200 shows a small 0.68% improvement over Rand with the added computational cost.

The first algorithm is the modification of the C-SCD by substituting the Ci pixel set for its overcompleted EM dictionary; the algorithm is then forced to select each member of the Ci to obtain the learned dictionary rather than by random selections. The pseudo-code of this K-Means SCD algorithm (KMSCD) is shown in algorithm 11, and the complexity of the KMSCD is $\approx O(J_c K^2)$, where (J_c) is the number of clusters. The complexity improvement is seen in the convergence curve presented in figure 5.15, showing that the KMSCD algorithm has converged between 2-3k iteration compared to the C-SCD around 8k iterations, which is around 2.6-3x improvement.

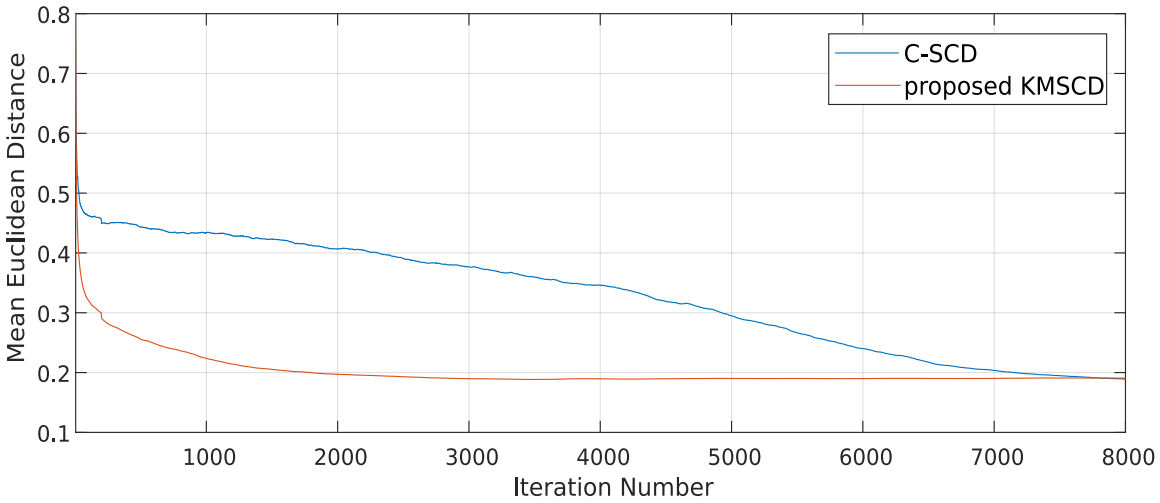


Figure 5.15: Convergence curve (with 399 point curve smoothing) trained on Selene H23 Dual scene shows that the proposed KMSCD has converged between 2-3k iterations compared to C-SCD at around 8k iterations.

Algorithm 11 Proposed K-Means SCD (KMSCD) algorithm.

1: **Import** HSI Image as 'Y', number of EMs 'm', user-defined step size 's' and step-size decay 'd', and maximum number of iterations as 'M'
2: Classify 'Y' into 'J_c' classes, where 'J_c' is also number of samples per iteration
3: **Initialise** EM dictionary as D_u as 'm' random numbers
4: **for** `k = 1 to K` **do**
5: Select Y_k samples, one from each 'J_c' class
6: $\min a_k$ in $||Y_k - D_u a_k||_F^2 = 0, \forall a_x \geq 0$ and $||a_k||_1 = 1$ ▷ Infer positive sum-to-one abundance
7: $D_k = (Y_k - D_u a_k) a_k^T$ ▷ EM update estimation
8: $D_u \leftarrow D_u + sD_k$ ▷ Update EM dictionary
9: s = s d ▷ Update step-size

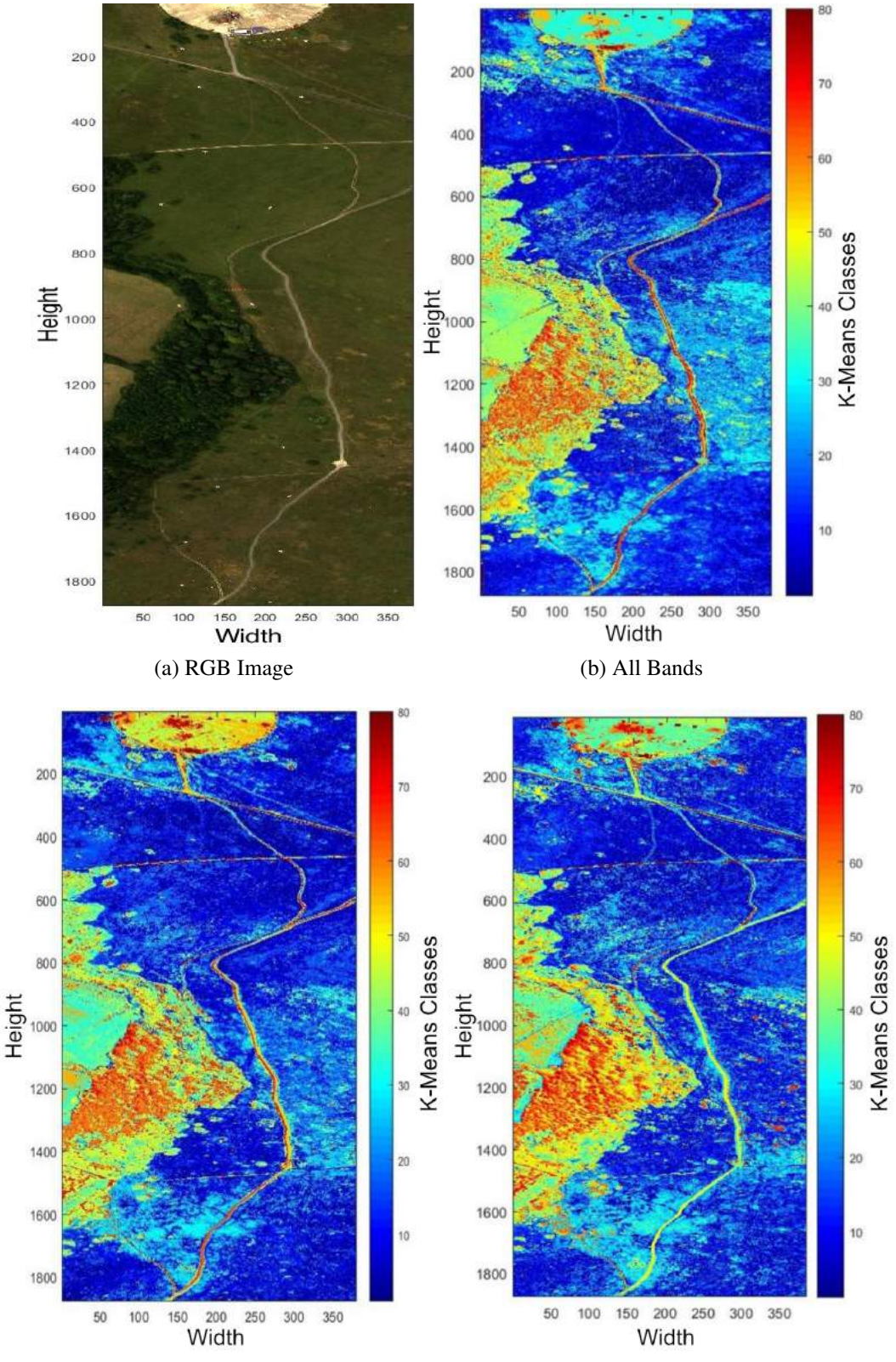
The second algorithm (Algorithm 12) is designed for applications where there is a limit on the maximum number of mixtures (N_{mp}) allowed in a given pixel. For the construction of pixel texture, a limit of up to three or four mixtures per pixel is permitted in most HSI scene simulators such as the CameoSim. [81]. This algorithm 12 learns the D_u using the sampling technique in the proposed KMSCD, but the abundance 'a' is subsequently constrained through the FNNOMP, which limits the maximum mixture per pixel to be four different materials (EMs). It is to be noted that, in Algorithm 11, there is no limit on the number of mixtures per pixel.

Algorithm 12 Proposed for scene simulators: KMSCD+FNNOMP.

1: **Import** HSI Image as 'Y', number of EMs 'm', maximum number of EM materials per pixel 'N_{mp}', user-defined step size 's' and step-size decay 'd', and maximum number of iterations as 'M'
2: Classify 'Y' into 'J_c' classes, where 'J_c' is also number of samples per iteration
3: **Initialise** EM dictionary as D_u as 'm' random numbers
4: **for** `k = 1 to K` **do**
5: Select Y_k samples, one from each 'J_c' class
6: a_x = FNNOMP(D_u, Y_k, N_{mp}), $\forall ||a_i||_1 = 1$
7: $D_k = (Y_k - D_u a_k) a_k^T$
8: $D_u \leftarrow D_u + sD_k$
9: s = s d

5.2.1 Feasibility of K-Means Clustering for Multispectral Data Set

As mentioned earlier, one of the aims of this thesis is to achieve an efficient reconstruction of the HSI scene, provided a multi-spectral image (MSI) input. The method proposed includes a pre-processing clustering method to extract the scene's spectral characteristics. This section aims to evaluate how robust the clustering is when only a few $\leq$ 10 spectral bands are present in input data. Figure 5.16 depicts the false-colour maps of the Selene Dual scene clustered by the unsupervised algorithm, K-Means, into arbitrarily selected 80 centres over 100 iterations. The input data sets consist of (i) all 448 hyperspectral bands, (ii) 16 WorldView-3 (WV3) centred wavelengths and (iii) 8 WorldView-2 (WV2) centred wavelengths. The pixel colour in the figure reflects the classes they are in and the colour assignment is random, which means that the colour of the ith class can be different in all three instances. The figure highlights how well these three data sets cluster grass, tree, soil, artificial materials, etc. over the very different number of spectral bands. The classified patterns are shown in figure 5.16 exhibit $\approx$99% similarity over the three results, indicating that the clustering method is robust against the number of spectral bands of the input MSI data demonstrating the practicability of the KMSCD for the reconstruction of HSI from the MSI imagery.



(a) RGB Image

(b) All Bands

(c) WV3

(d) WV2

Figure 5.16: False-colour classification maps displayed with (b) all 448 bands of the H23 Dual image, (c) with the 16 centre wavelengths of the WV3 sensor, and (d) with 8 centre wavelengths of the WV2 sensor classified by K-Means into 80 classes with MATLAB's default maximum of 100 iterations. The figures show $\geq$99% of classification similarities despite of the very small number of spectral bands (8 bands) that has been utilised in (d). [3]

5.2.2 C-SCD vs. KMSCD: Reconstruction of Background Pixels

Robustness of C-SCD and the Proposed KMSCD

This section aims to illustrate the robustness of the dictionary learning (DL) in C-SCD [11] and to compare this with the proposed KMSCD method (Algorithm 11). The robustness can be tested by examining whether reproducibility can be found on repeated runs of both algorithms over the most abundant EM in the scene. In both cases the experiments were conducted using the entire data set for DL and the K-Means cluster number (Jc) set to the number of samples selected per iteration, which was 200 in C-SCD paper [11]. Two experiments were performed to test the algorithms. In the first experiment, the number of dictionary atoms were gradually increased starting with 10 atoms to 50 atoms at an increment of 10 (i.e. 10 atoms, 20 atoms, ..., 50 atoms). The second experiment is to test for the reproducibility of the results, performing five repeated runs under the same training conditions with 40 dictionary atoms. Both experiments have shown the superior performance of the KMSCD dictionary, and figure 5.17 shows the mean DL1NE error with increase in the number of dictionary atoms.

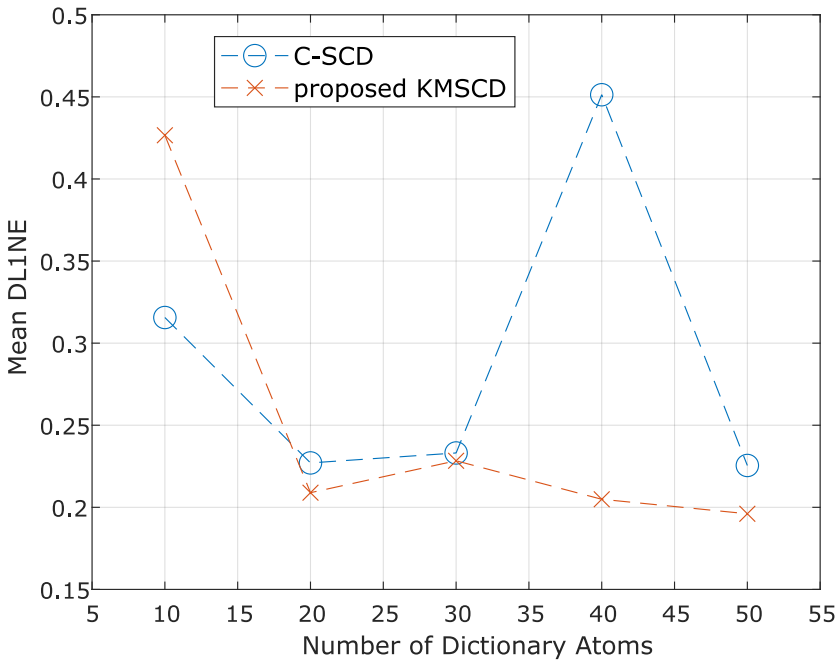
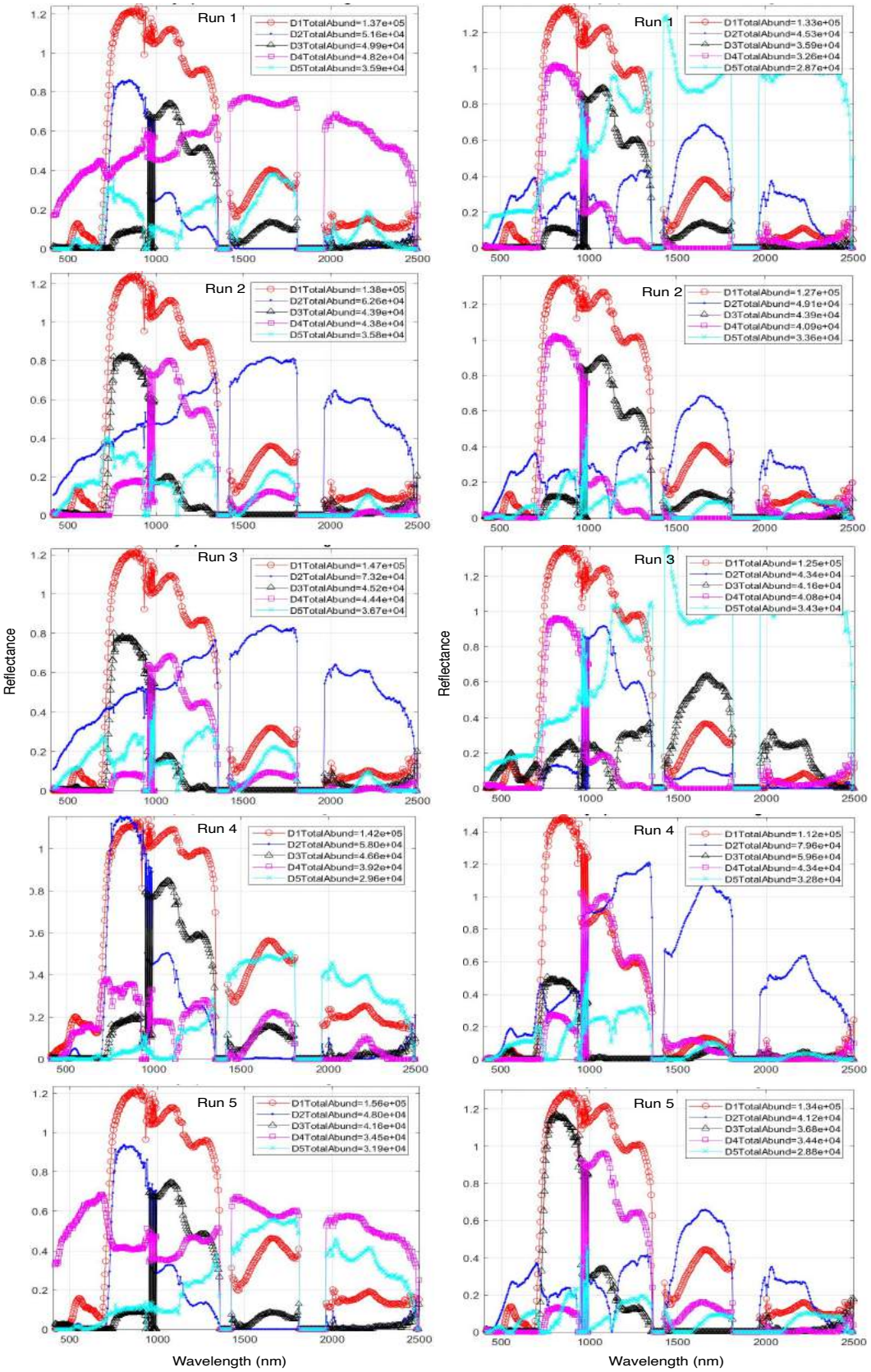


Figure 5.17: Mean DL1NE error against increasing number of dictionary atoms. The result shows that the proposed KMSCD has a ≈13% mean improvement over C-SCD.

Similar improvement in the reconstruction accuracy is seen with EM dictionary (which contains sum-to-one abundance). The figure 5.18 represents the spectra of the first five most abundant materials in the reconstructed Selene Dual data set over five repeated runs that the C-SCD and the KMSCD algorithm have learned. The colour of the line plot is fixed as according to the order of the abundances, e.g., the red and the blue plots indicate the most and second most abundant materials in the reconstructed scene, respectively. Both methods manage to find the most abundant materials (in red plot) in four out of the five runs (i.e., they fail to find the most abundant materials correctly in the fourth run) with the total abundance of this material in the order of $\approx 1.4 \times 10^5$, which is roughly $\approx$20% of the scene. The scene contains $1876 \times 380 \approx 7.1 \times 10^5$ pixels. However, the second most abundant material (in blue plot) that has been found by the C-SCD exhibit two quite distinct EMs with total abundances ranging between 5×10^4 to 7×10^4 respectively; whereas the KMSCD gives three out of five runs the same EM with abundances of $\approx 4.4 \times 10^4$, which amounts to $\approx$6% of the scene. Therefore this result may imply that the proposed KMSCD performs slightly more robustly than the C-SCD algorithm.



(a) C-SCD unmixing (b) KMSCD unmixing

Figure 5.18: Most abundant endmembers (EMs) for the five runs with 40 EMs between (a) SCD-unmixing with random sample selection and (b) the proposed K-Means SCD algorithm (KMSCD) unmixing. [3]

The robustness of the C-SCD and the KMSCD algorithms is further examined through their reproducibility over the repeated five runs of scene reconstructions. Figure 5.19 plots the mean of the DL1NE error given by the C-SCD and KMSCD algorithms. Over the five runs, the mean errors and standard deviations (std) of the C-SCD and KMSCD are, respectively, $0.32\% \pm 0.057\%$ and $0.24\% \pm 0.027\%$. The KMSCD's standard deviation (std) over the five runs exhibit approximately half of that by C-SCD, further demonstrating the much greater robustness of the proposed KMSCD algorithm compared to the C-SCD.

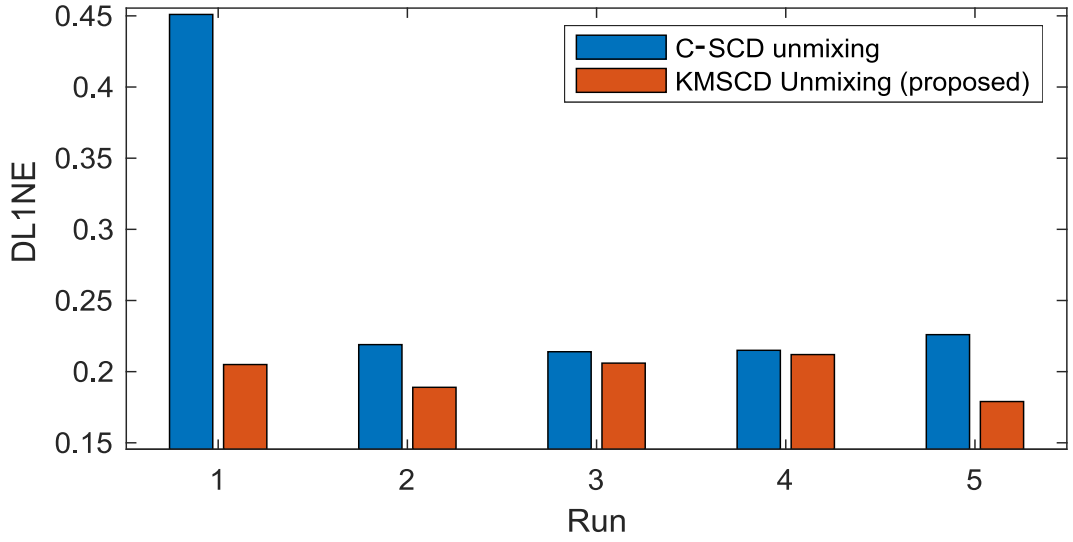
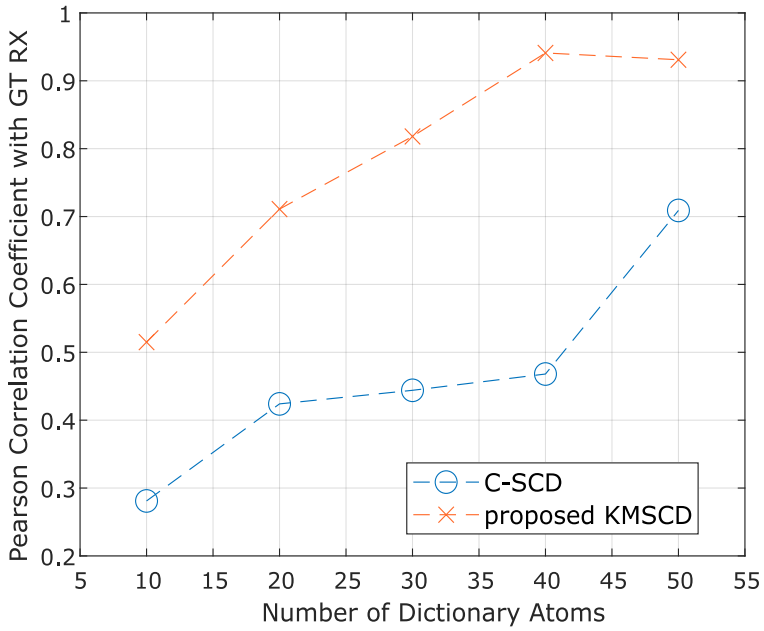
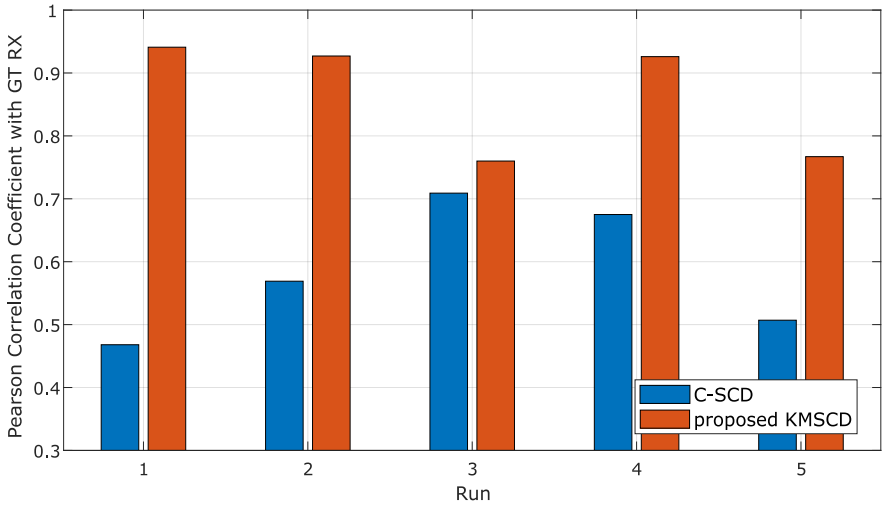


Figure 5.19: Plots the mean of the differential L1 norm (DL1NE) of the 5 repeated runs of the Selene Dual scene reconstruction performed by the C-SCD and the proposed KMSCD DL learning algorithms. The STD of the DL1NE processed by the C-SCD is almost double of that processed by the proposed method over the 5 experimental runs, further demonstrating the superior performance of the proposed KMSCD algorithm. [3]

Correlation comparing the RX detection of the reconstructed data with the ground truth (GT) in figure 5.20 show a $\approx$69.6% mean improvement with the increase in the number of dictionary atoms and $\approx$47.4% mean improvement on five repeated runs with the same training conditions. EM dictionary (with sum-to-one abundance condition) sees a similar improvement of $\approx$68.5% with increase in the number of endmembers and $\approx$73.4% in the five runs under the same training conditions. The numerical accuracy of the results is presented in appendix B.1.



(a) RX detection accuracy with increasing number of dictionary atoms. KMSCD show a ≈69.6% mean improvement compared to C-SCD.

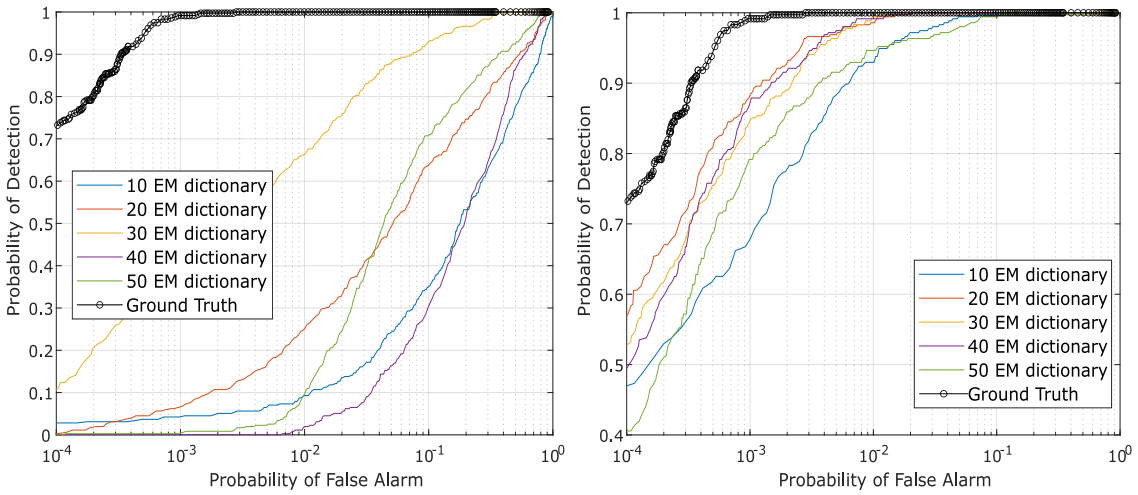


(b) Five repeated runs with 40 dictionary atoms. KMSCD show a ≈67% mean improvement compared to C-SCD.

Figure 5.20: Pearson correlation coefficient between the GT RX and the RX of the reconstructed image performed by C-SCD and the proposed KMSCD algorithms. The figure (a) shows the accuracy with increasing number of dictionary atoms, and (b) shows five runs under the same training conditions.

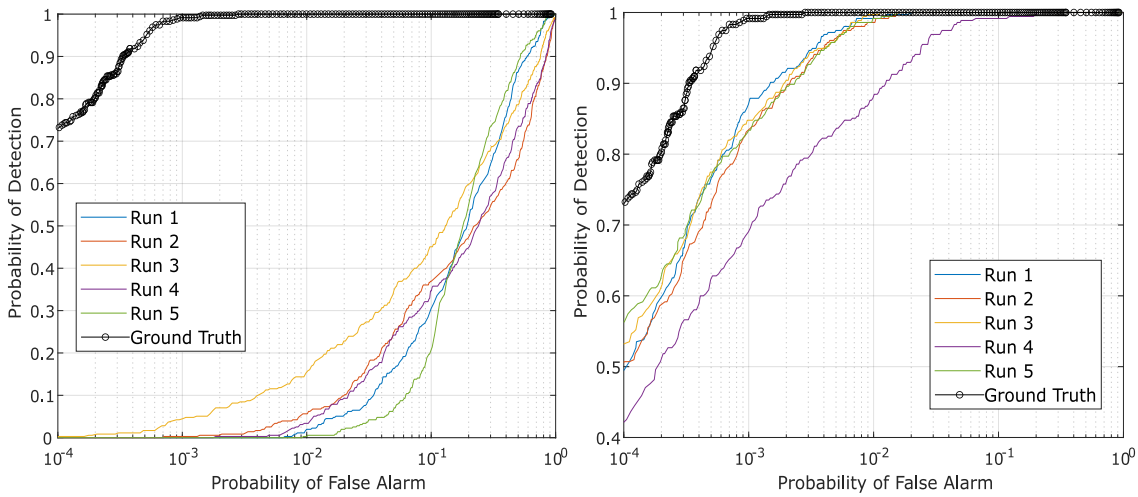
Similarly, target detection performed with Orange Perspex target material using the ACE algorithm show a ≈20.4% mean improvement in AUC with increasing number of EMs, which is presented in figure 5.21. And, ≈40% mean improvement in AUC was achieved on the five runs under the same training parameters with 40 EMs. The ROC

curve is of five runs is presented in figure 5.22, with detailed numerical accuracy in appendix B.2.



(a) C-SCD (b) proposed KMSCD

Figure 5.21: Receiver operating characteristics (ROC) curve of ACE detection for Orange Perspex material with increase in the number of EMs. Results show a ≈20.4% improvement in the mean and ≈5x in standard deviation over C-SCD unmixing.



(a) C-SCD (b) proposed KMSCD

Figure 5.22: ROC curve of five repeated runs with 40 EMs. Results show that the proposed KMSCD unmixing has a ≈40% mean improvement in AUC with a ≈525x improvement in the standard deviation over C-SCD unmixing.

Accuracy of C-SCD and KMSCD: Background Pixels

This section presents the main results of this thesis and the first focus is to verify the effectiveness of the reconstruction of the scene for the scene's background material. The capability of the DL algorithms to reconstruct trace materials is subsequently presented in the next section. Figure 5.24 shows the DL1NE (i.e., the differential ℓ_1-norm of the reconstructed vector w.r.t. that of the ground truth) false-colour map of the Selene Dual scene reconstructed by the proposed KMSCD (i.e., the Algorithm 11) and also by other competing algorithms. n all cases the data input was the Selene Dual 448 band data set, and the dictionary sizes to be learned were set at M = 50. Ideally, a cluster is considered to be a candidate substance by itself in the classification.

The random selection of samples in C-SCD is substituted and KMSCD is coerced into choosing one sample from each cluster. The DL was implemented using the first 1000 lines (about half of the scene) for all methods. The result of the reconstruction presented in Figure 5.24 is in false colour, and, in order to illustrate the quality of the reconstruction output across the entire scene among all competing approaches, all results were presented showing errors up to a maximum of 3 times the DL1NE error mean across the entire scene. For example, the colour patterns in figure 5.24a, b are very similar: both show high (red) errors in the tree areas, while good reconstructions (i.e., blue) are seen over the glass and concrete slabs in the scene. The differences between these two are that the proposed KMSCD result (Figure 5.24a) gives an average of 0.28% of the DL1NE over the entire scene, which is $\approx$40% better than that of the C-SCD (Figure 5.24b) for reconstruction. These two figures also highlight the higher superiority in the proposed KMSCD for the reconstruction of the scene's most abundant material (i.e., grass) over the C-SCD method: the entire scene apart from the tree area is all in very low error $<$0.2% (in blue colour) and can only be seen from the KMSCD outcome.

The competing methods of unmixing (as stated in Chapter 2 of this thesis) are a mixture of the classical unmixing method assuming a simplex and statistical form using mathematical optimisation methods. Classical methods are the popular VCA and the recent SD-SOMP algorithm (with pure pixel assumption) and MVSA algorithm (without

pure pixel assumption), and statistical form of unmixing methods are the C-SCD and the CoNMF algorithm. Table 5.2 shows that, over the five data sets used in this analysis, the proposed KMSCD is capable of reconstructing certain scenes with the lowest error over all competing algorithms adopted in this research. Similar to Table 5.3, Table 5.4 tabulates the mean of the entire scene's MD error for various data sets used in this research. Table 5.4 is the same as Table 5.3, but the mean(MD/band) has been tabulated so to allow direct comparisons between data sets that may have a differing number of bands.

From these tables it is seen that the smallest errors (DL1NE and MD) in all five data sets used in this study are achieved through the proposed KMSCD method, which shows performances that are about 20–500 percent better than all competing algorithms. The enhancement figures that are tabulated at the bottom line of Tables 5.2–5.4, which utilise two different error metrics (i.e., the DL1NE and the MD) for the assessment, implying that the accuracy assessments are highly consistent. The best performances are denoted in bold.

Table 5.2: Mean DL1NE showing the reconstruction error with various HSI images. Results show a ≈20.6-530% improvement in reconstruction accuracy using the proposed KMSCD method.

Hyperspectral Images	**Proposed**	**C-SCD Unmix**	**SD-SOMP**	**CoNMF**	**MVSA**	**VCA**
Selene H23 VNIR	**1.25**	1.60	2.54	2.33	2.99	2.34
Selene H23 Dual	**0.28**	0.41	1.38	0.61	0.99	1.32
Paso Robles-Monterey	1.23	1.22	4.27	**0.77**	9.04	9.04
Virginia City 1807-1220	**0.05**	0.11	0.86	0.16	2.57	2.57
Virginia City 1807-1259	**0.06**	0.13	1.06	0.17	2.83	2.83
Mean error ± Std	**0.57** ±**0.61**	0.69 ±0.68	2.02 ±1.42	0.81 ±0.89	3.68 ±3.10	3.62 ±3.08
Enhanced reconstruction accuracy over 5 datasets w.r.t. KMSCD		20.64%	251.79%	40.24%	540.93%	529.90%

First 1000 lines is used for training the dictionary with 50 atoms each for scenes Selene H23 VNIR, Selene H23 Dual and Paso Robles-Monterey. Reconstructed Virginia City images were trained on Virginia City 1807-1211.

Table 5.3: Mean Manhattan distance reconstruction error on the HSI images. Results show a ≈7-318% improvement of the proposed KMSCD over other methods.

Hyperspectral Images	Proposed	SCD Unmix	SD-SOMP	CoNMF	MVSA	VCA
Selene H23 VNIR	**1.47**	1.60	2.65	2.29	2.77	2.83
Selene H23 Dual	**2.33**	2.51	3.94	2.69	3.63	5.46
Paso Robles-Monterey	**1.93**	1.99	7.37	**1.93**	15.79	15.79
Virginia City 1807-1220	**0.25**	0.30	0.94	**0.25**	0.84	0.99
Virginia City 1807-1259	**0.25**	0.28	0.98	**0.25**	0.82	0.99
Mean error ± Std	**1.24** ±**0.96**	1.34 ±1.01	3.18 ±2.66	1.48 ±1.16	4.77 ±6.28	5.21 ±6.19
Enhanced reconstruction accuracy over 5 datasets w.r.t. KMSCD		7.22%	154.90%	18.94%	282.83%	318.40%

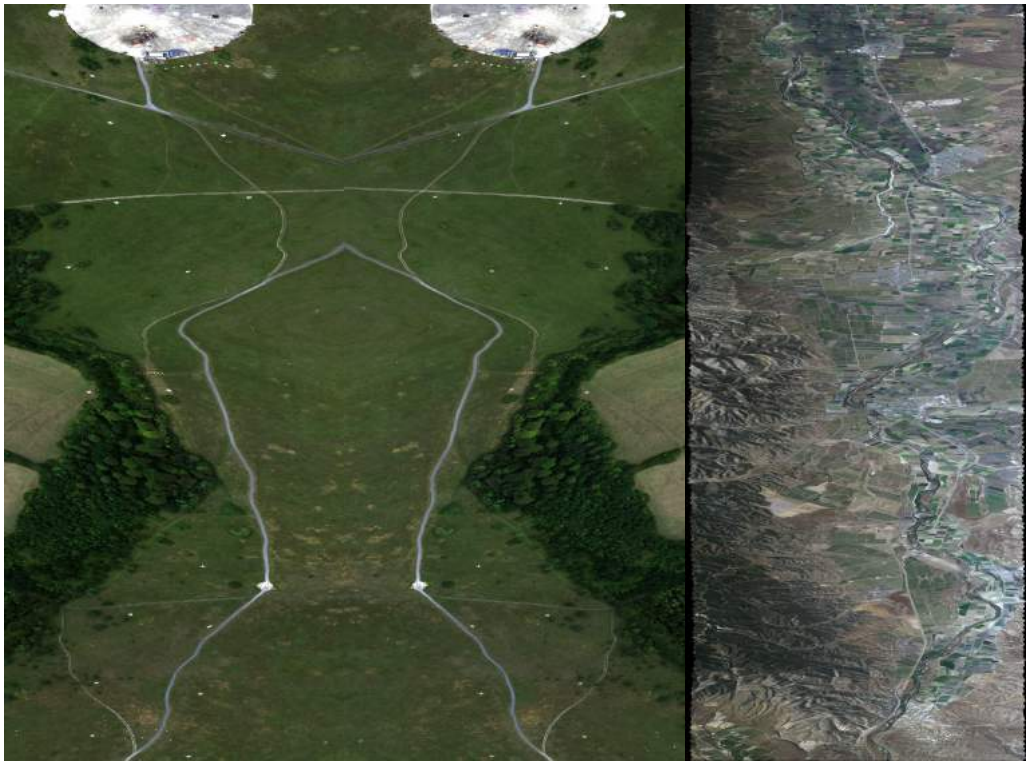
First 1000 lines is used for training the dictionary with 50 atoms each for scenes Selene H23 VNIR, Selene H23 Dual and Paso Robles-Monterey. Reconstructed Virginia City images were trained on Virginia City 1807-1211.

Table 5.4: Mean Manhattan distance reconstruction error/band of the HSI scenes. Results show that the proposed KMSCD is ≈7-330% better.

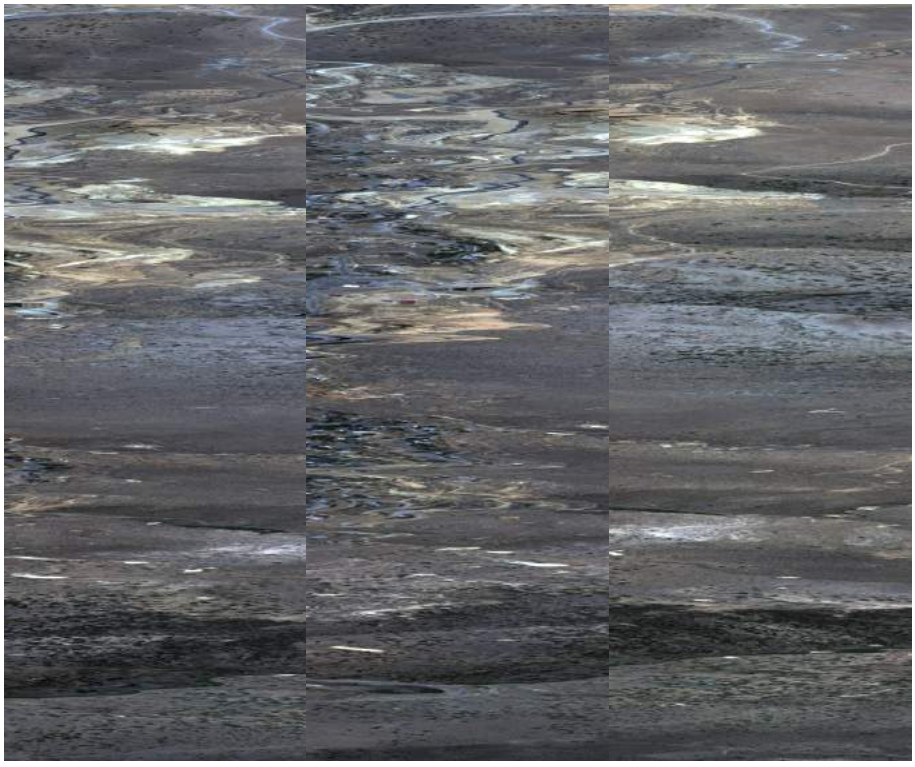
Hyperspectral Images	Proposed	SCD Unmix	SD-SOMP	CoNMF	MVSA	VCA
Selene H23 VNIR	**9.2e-03**	1.0e-03	1.66e-02	1.43e-02	1.73e-02	1.77e-02
Selene H23 Dual	**5.2e-03**	5.6e-03	8.8e-02	6.0e-03	8.1e-03	1.22e-02
Paso Robles-Monterey	**8.6e-03**	8.9e-03	3.29e-02	**8.6e-03**	7.05e-02	7.05e-02
Virginia City 1807-1220	**1.4e-03**	1.7e-03	5.3e-03	**1.4e-03**	4.7e-03	5.6e-03
Virginia City 1807-1259	**1.4e-03**	1.6e-03	5.5e-03	**1.4e-03**	4.6e-03	5.6e-03
Mean error ± Std	**5.2e-03** ±**3.8e-03**	5.6e-03 ±3.9e-03	1.38e-02 ±1.16e-02	6.3e-03 ±5.4e-03	2.1e-02 ±2.81e-02	2.23e-02 ±2.74e-02
Enhanced reconstruction accuracy over 5 datasets w.r.t. KMSCD		7.75%	167.83%	22.87%	307.75%	332.56%

First 1000 lines is used for training the dictionary with 50 atoms each for scenes Selene H23 VNIR, Selene H23 Dual and Paso Robles-Monterey. Reconstructed Virginia City images were trained on Virginia City 1807-1211.

The RGB images of all six hyperspectral images are presented in figure 5.23, and the DL1NE error maps of all reconstructed scenes are presented from figure 5.24 to figure 5.28.

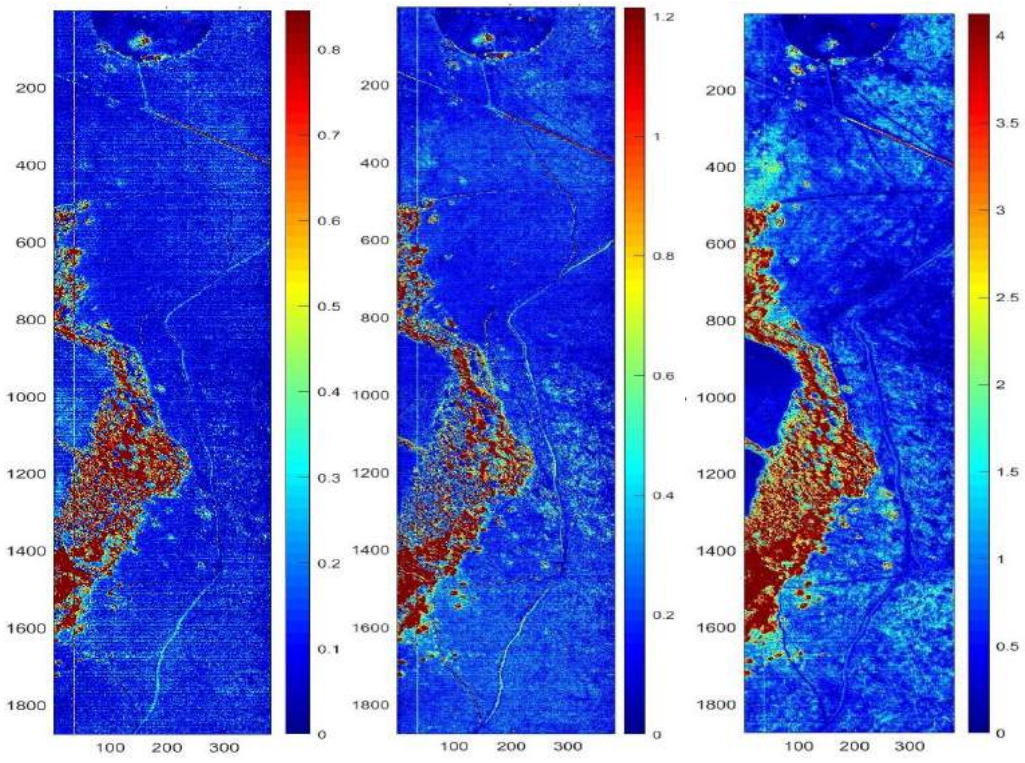


(a) Selene H23 Dual (b) Selene H23 VNIR (c) Paso Robles-Monterey

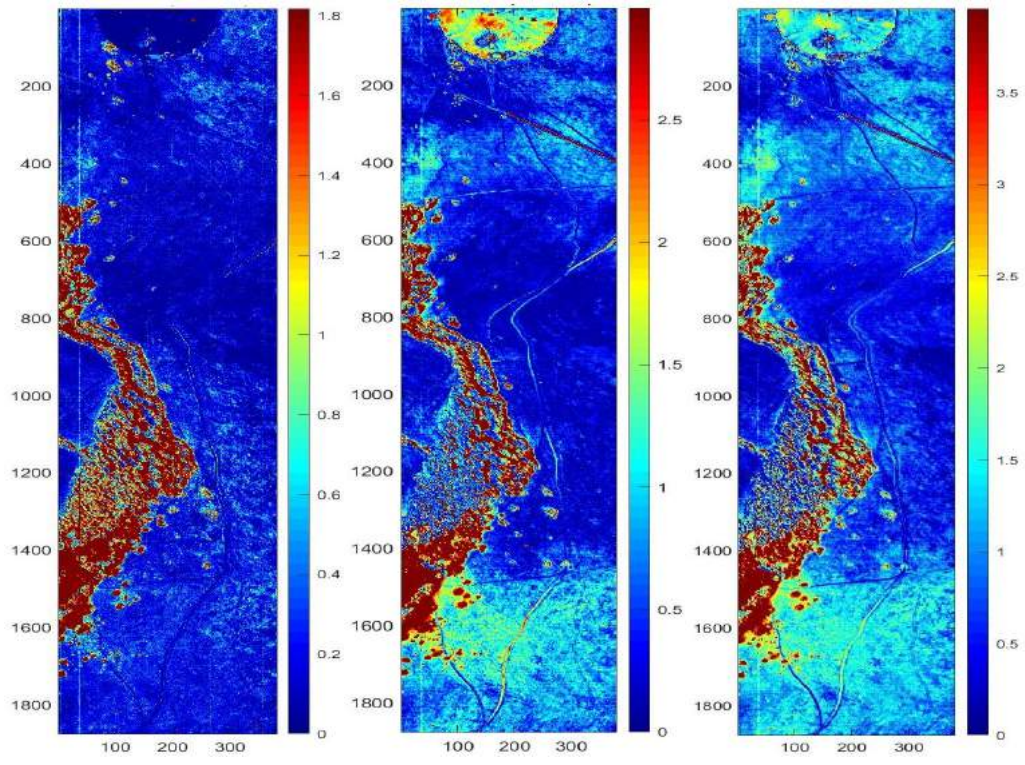


(d) VC 1807-1211 (e) VC 1807-1220 (f) VC 1807-1259

Figure 5.23: RGB images of the different datasets used in this thesis: Selene H23 VNIR and Dual, Paso Robles-Monterey, and three Virginia City (VC) datasets.

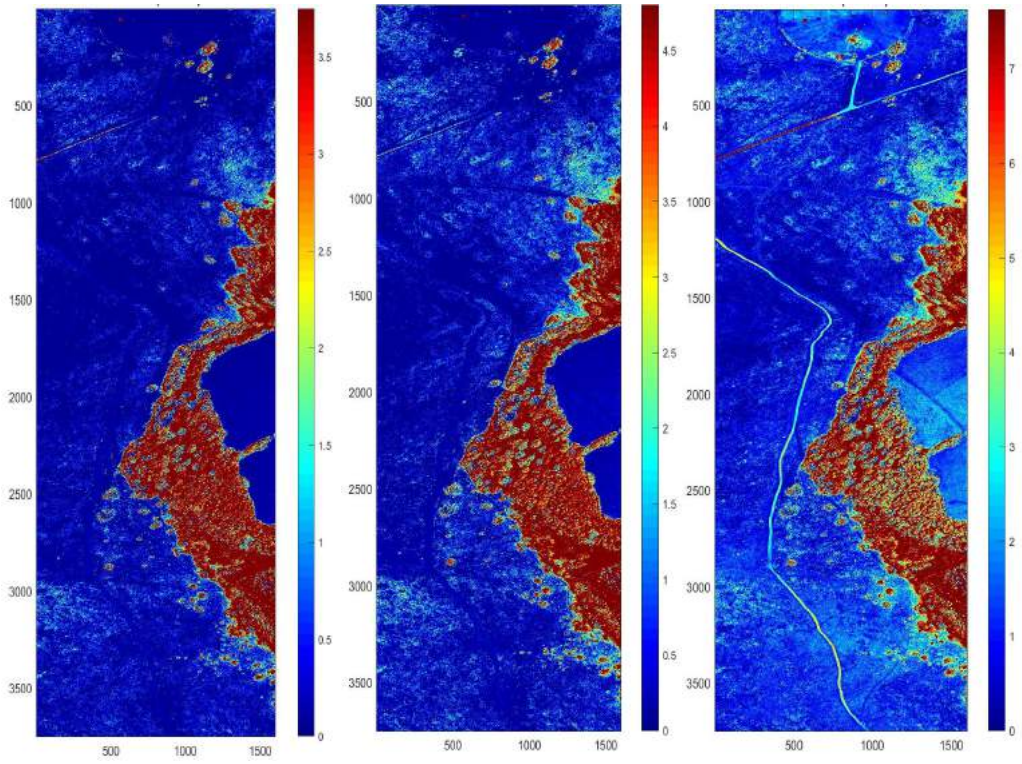


(a) KMSCD Unmix (b) C-SCD Unmix (c) SD-SOMP

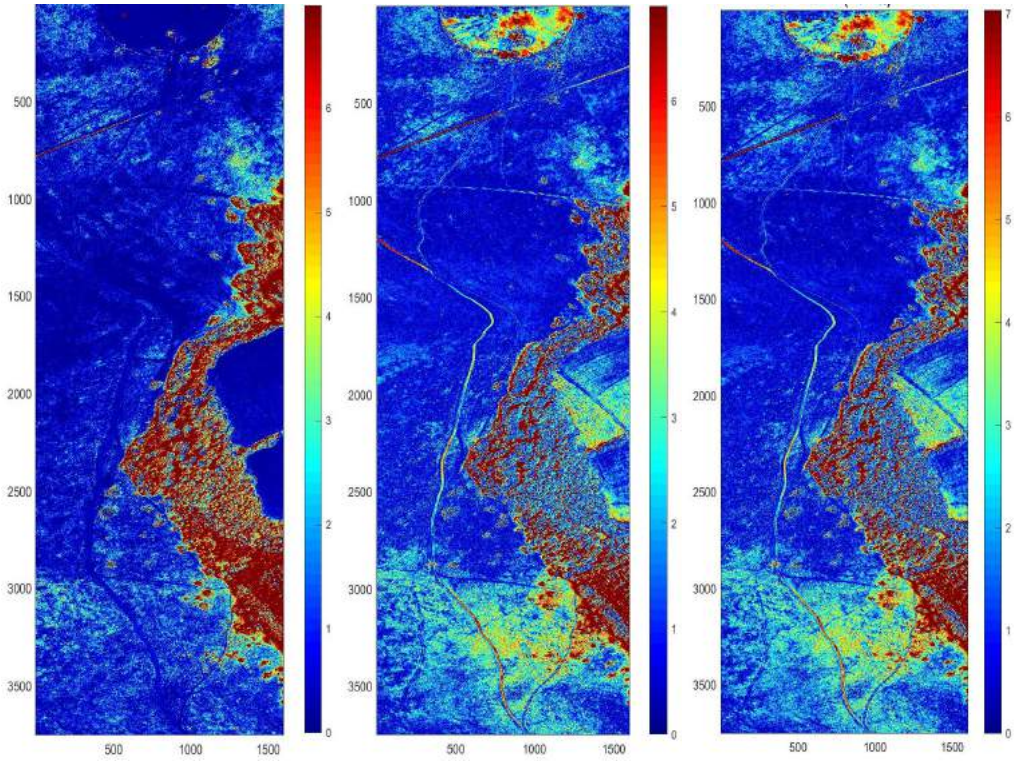


(d) CoNMF (e) MVSA (f) VCA

Figure 5.24: False-colour map of DL1NE of different methods on the reflectance data of H23 Dual scene when trained from the first 1000 lines, whose mean error is mentioned in Table 5.2. Each of the error maps has been presented in various scales of [0 to 3 $\times$ mean(DL1NE)], such that the consistency of the reconstruction performance over the entire scene among all methods can be examined. [3]

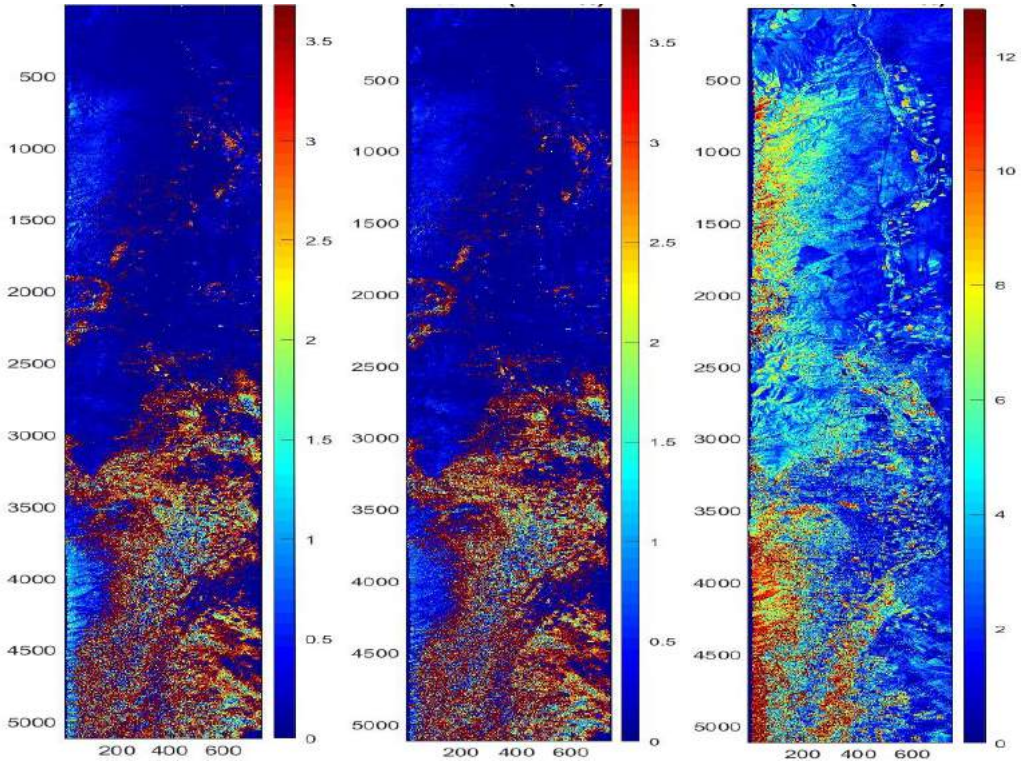


(a) KMSCD Unmix (b) C-SCD Unmix (c) SD-SOMP

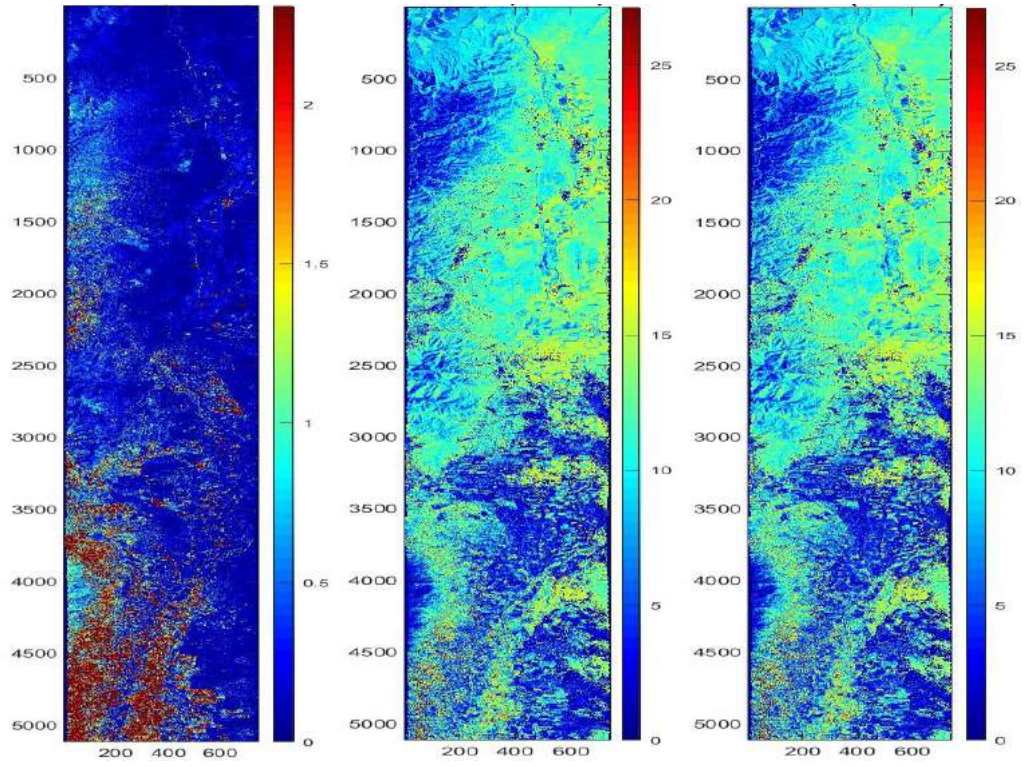


(d) CoNMF (e) MVSA (f) VCA

Figure 5.25: False-colour map of DL1NE of different methods on the reflectance data of H23 VNIR scene when trained from the first 1000 lines, whose mean error is mentioned in Table 5.2. Each of the error maps has been presented in various scales of [0 to 3 $\times$ mean(DL1NE)], such that the consistency of the reconstruction performance over the entire scene among all methods can be examined.

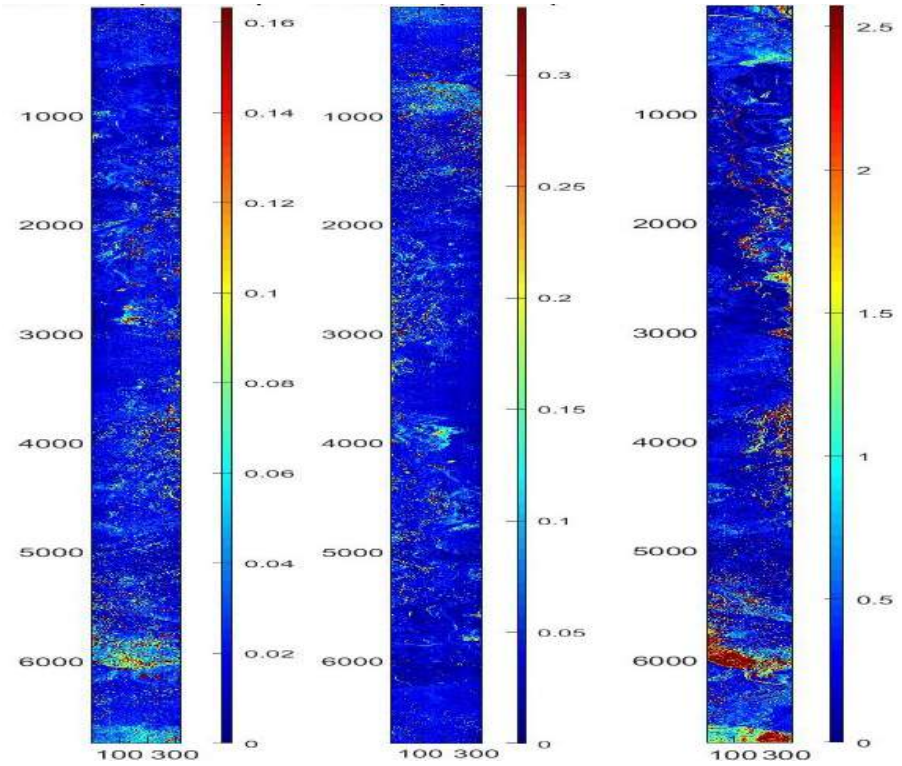


(a) KMSCD Unmix (b) C-SCD Unmix (c) SD-SOMP

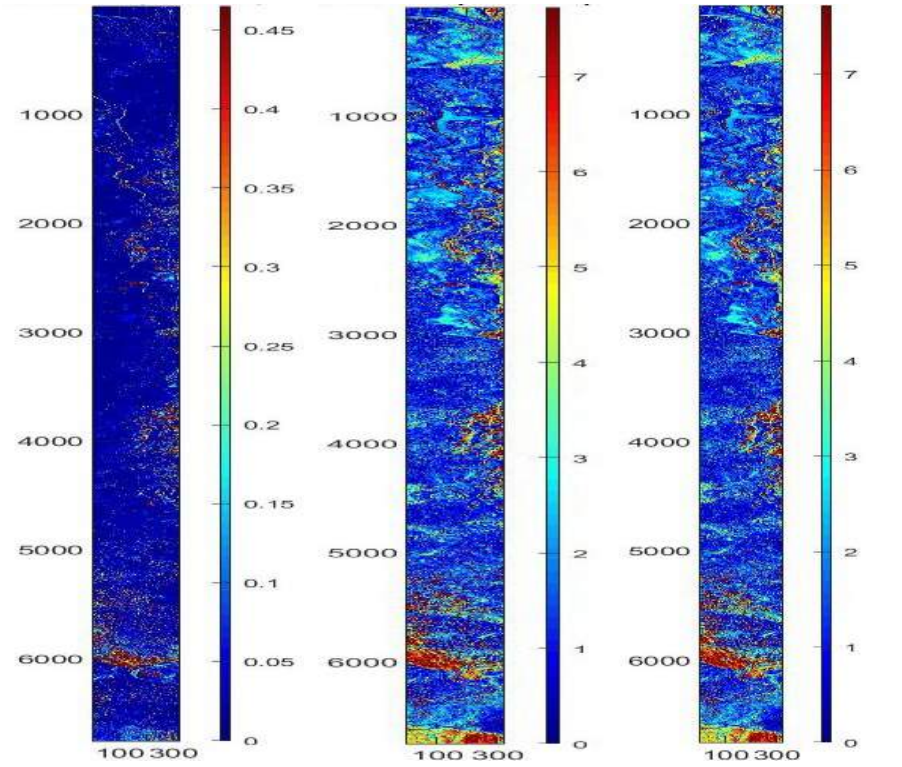


(d) CoNMF (e) MVSA (f) VCA

Figure 5.26: False-colour map of DL1NE of different methods on the reflectance data of Paso Robles-Monterey scene when trained from the first 1000 lines, whose mean error is mentioned in Table 5.2. Each of the error maps has been presented in various scales of [0 to 3 $\times$ mean(DL1NE)], such that the consistency of the reconstruction performance over the entire scene among all methods can be examined.

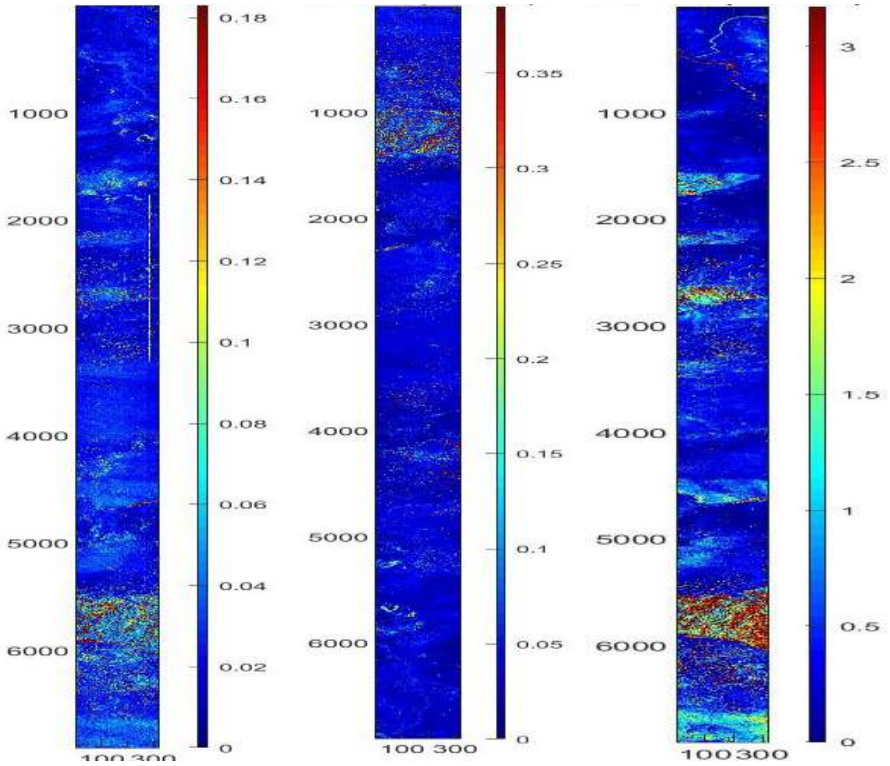


(a) KMSCD Unmix (b) C-SCD Unmix (c) SD-SOMP



(d) CoNMF (e) MVSA (f) VCA

Figure 5.27: False-colour map of DL1NE of different methods on the reflectance data of Virginia City (VC) 0817-1220 scene when trained from the VC 0817-1211 scene, whose mean error is mentioned in Table 5.2. Each of the error maps has been presented in various scales of [0 to 3 $\times$ mean(DL1NE)], such that the consistency of the reconstruction performance over the entire scene among all methods can be examined.



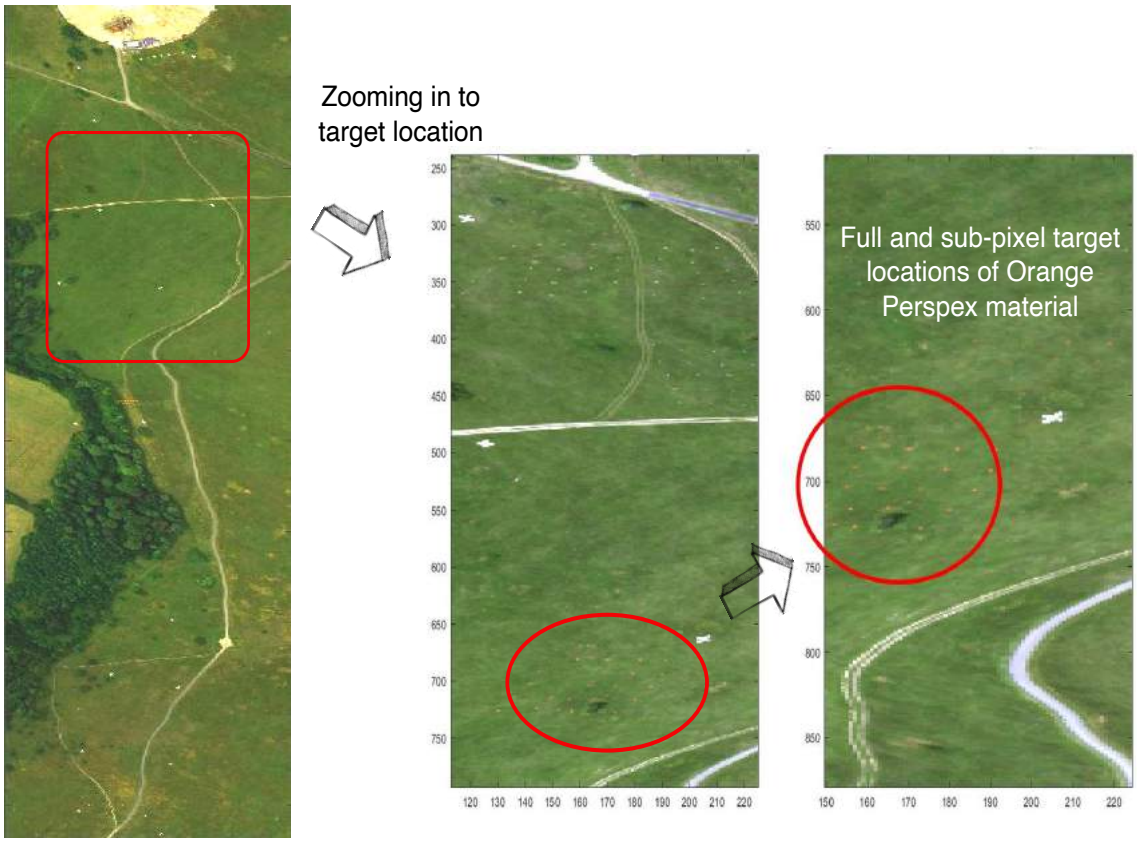
(a) KMSCD Unmix (b) C-SCD Unmix (c) SD-SOMP

(d) CoNMF (e) MVSA (f) VCA

Figure 5.28: False-colour map of DL1NE of different methods on the reflectance data of Virginia City (VC) 0817-1259 scene when trained from the VC 0817-1211 scene, whose mean error is mentioned in Table 5.2. Each of the error maps has been presented in various scales of [0 to 3 $\times$ mean(DL1NE)], such that the consistency of the reconstruction performance over the entire scene among all methods can be examined.

C-SCD vs. KMSCD

This section describes the capabilities of the DL algorithms to reconstruct trace materials in the scene, specifically, the C-SCD and the proposed KMSCD algorithms. There is a small amount (≈1 of whole scene) of full-pixel (i.e., material occupancy = 1) artificial materials embedded in the Selene scene, and their presence in the scene are depicted in the RGB image as shown in Figure 5.29a. A number of small Orange Perspex panels with dimensions of around 40 × 40 cm, deployed as targets in the scene, are used here to testify to the ability of the DL algorithms to recover. Due to the random selection of the pixel strategy adopted for the DL in the C-SCD algorithm, the C-SCD's recovery of the Orange Perspex targets is seen to fail, as shown in Figure 5.29b. In contrast, these Orange Perspex targets have been successfully recovered by the proposed KMSCD, as shown in Figure 5.29c.



(a) Full and subpixel Orange Perspex target locations in ground truth.

Figure 5.29: Continued on the next page.

(c) C-SCD unmixing with random sample selection showing white patch at target location because the EM is not learned.

(d) Proposed KMSCD unmixing with similar RGB as ground truth image

Figure 5.29: RGB image of the Selene Dual data set showing the position of the target materials and DL algorithms' ability to recover them. [3]

Target detection with ACE

In order to measure the ability of the DL algorithms to obtain these small targets, the receiver operating characteristic (ROC) for detecting the Orange Perspex from the Selene Dual scene, reconstructed by C-SCD and the KMSCD for the same run as previously shown with 50 EMs, are shown in Figure 5.30. From the ROC curve it is quite clear that the KMSCD has detected the Orange Perspex targets ≈12 percent better than those from the C-SCD results, as according to the conventional method of assessment using the aura under the curve (AUC) metric to quantitatively evaluate the detection capability.

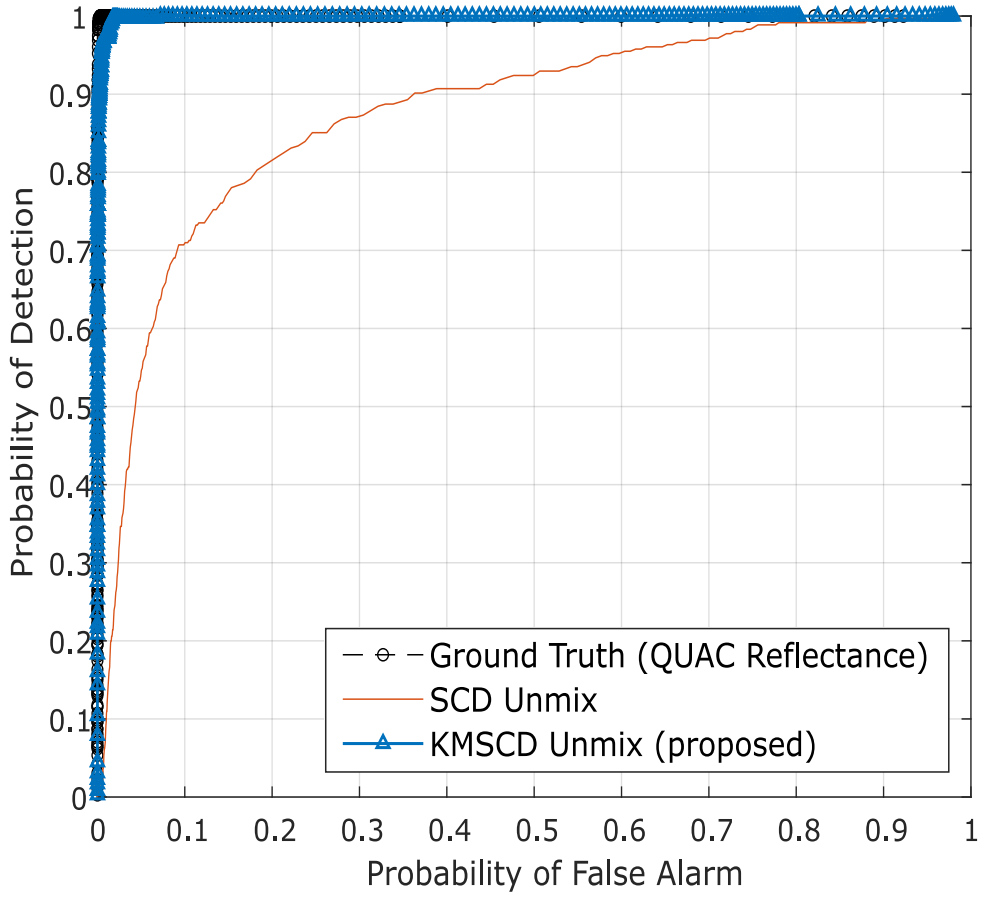
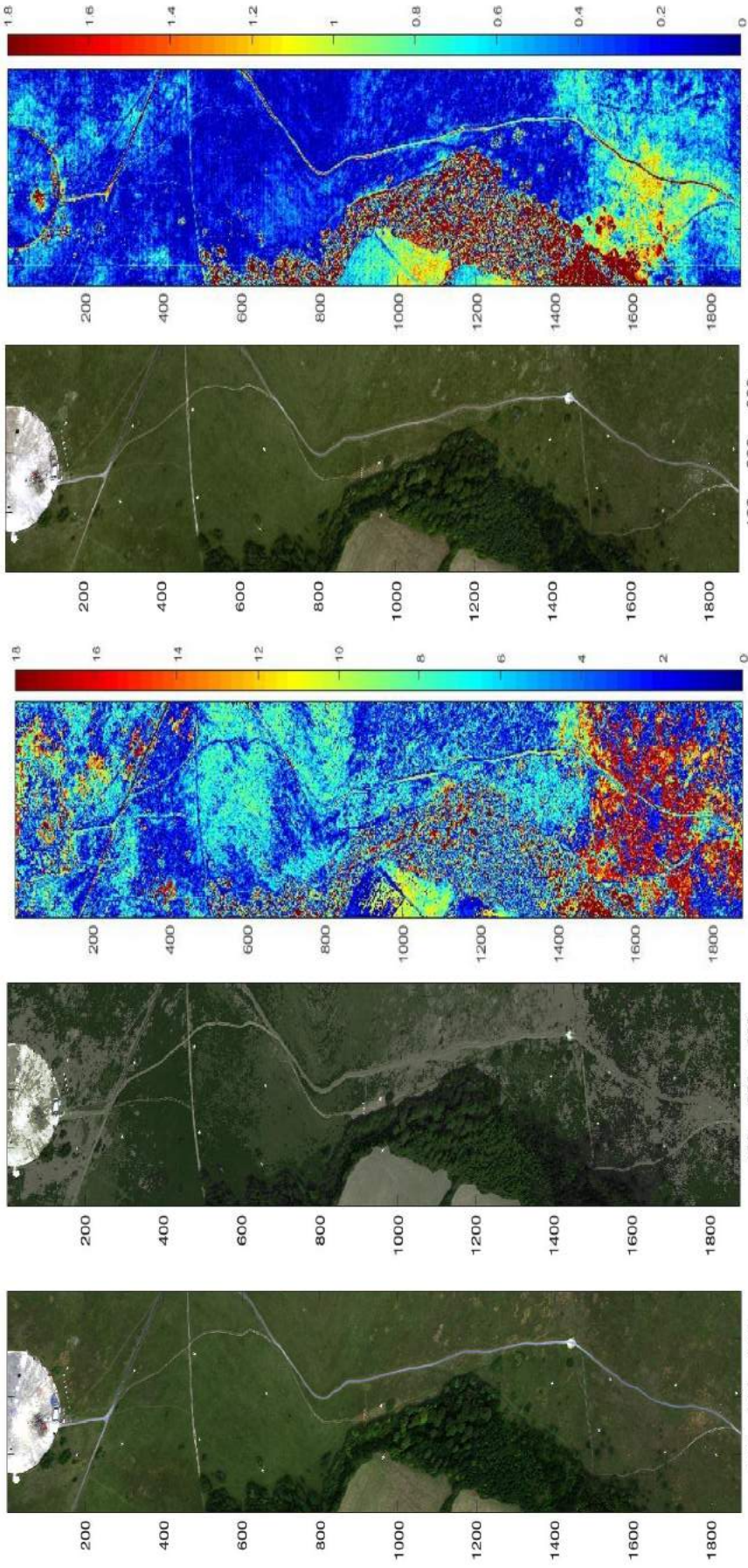


Figure 5.30: The receiver operating characteristic (ROC) for detecting the Orange Perspex target material in the Selene H23 Dual scene reconstructed by the algorithms C-SCD and KMSCD. The small orange targets are seen to be ≈12% better detected from the KMSCD reconstructed scene. [3]

5.3 Material allocation with KMSCD

Most HSI scene simulators, such as the commercial-off-the-shelf CameoSim package [17], enforce a constrain on the amount of materials (N_{mp}) that can coexist within each pixel in the scene, shown in figure 3.11. The result of this restriction is that it may affect the accuracy of HSI reconstruction, and it is this section's objective to try to determine the side effect of this constraint. Two different ways to constraint the N_{mp} have been implemented here: one is the use of the FNNOMP (i.e., the algorithm 12, KMSCD+FNNOMP, and the other uses the Texture Material Mapper (TMM) method (i.e., the KMSCD+TMM).

The TMM technique [81] was widely used in the HSI simulator (e.g., the CameoSim package) [82, 83, 84], which estimates the abundance by measuring the inverse of the Euclidean distances of each EM in relation to the spectral characteristics of the mixed pixel, i.e. the test pixel. Note that the results that have been presented in Section 5.2.2 are the reconstructions that have been processed without the limitation on the N_{mp}. Figure 5.31 depicts the false-colour DL1NE map of the Selene Dual scene reconstructed by Algorithm 12 (Figure 5.31e) and the TMM (Figure 5.31c) using a maximum of four materials (N_{mp}) in every pixel. The mean DL1NE error of the whole scene for the reconstruction by Algorithm 12 is found to be 0.74%, which is more than 2 times higher than that of the Algorithm 11 (i.e., the KMSCD without N_{mp} restriction). Furthermore, the mean DL1NE of the reconstruction that constrained by the TMM exhibits 7.12% error, which is $\approx$10 times higher than that of Algorithm 12.



(a) RGB GT (b) RGB (KMSCD+TMM) (c) DL1NE (KMSCD+TMM) (d) RGB (KMSCD+FNNOMP) (e) DL1NE (KMSCD+FNNOMP)

Figure 5.31: Selene H23 Dual scene where (a) is the ground truth RGB image, (b) and (c) are the reconstructed RGB and DL1NE error map with KMSCD+TMM, and (d) and (e) are the reconstructed RGB and DL1NE error map with KMSCD+FNNOMP (Algorithm 12). The mean errors of the entire map for panels (c,e) are 7.12% and 0.74%, respectively, showing the superiority of the FNNOMP over the TMM for constraining N_{mp} to four materials per pixel. [3]

Figure 5.32 plots the ROC for the detection by the ACE detector of the Orange Perspex target from these two scenes, and the AUC for the Algorithm 12 shows approximately twice that of the one limited by TMM. The combined results of these two figures that suggest that the proposed DL method using KMSCD and, together with the FNNOMP, will be more suitable as the material allocation method for realistic HSI scene simulation applications for HSI scene simulators such as the CameoSim.

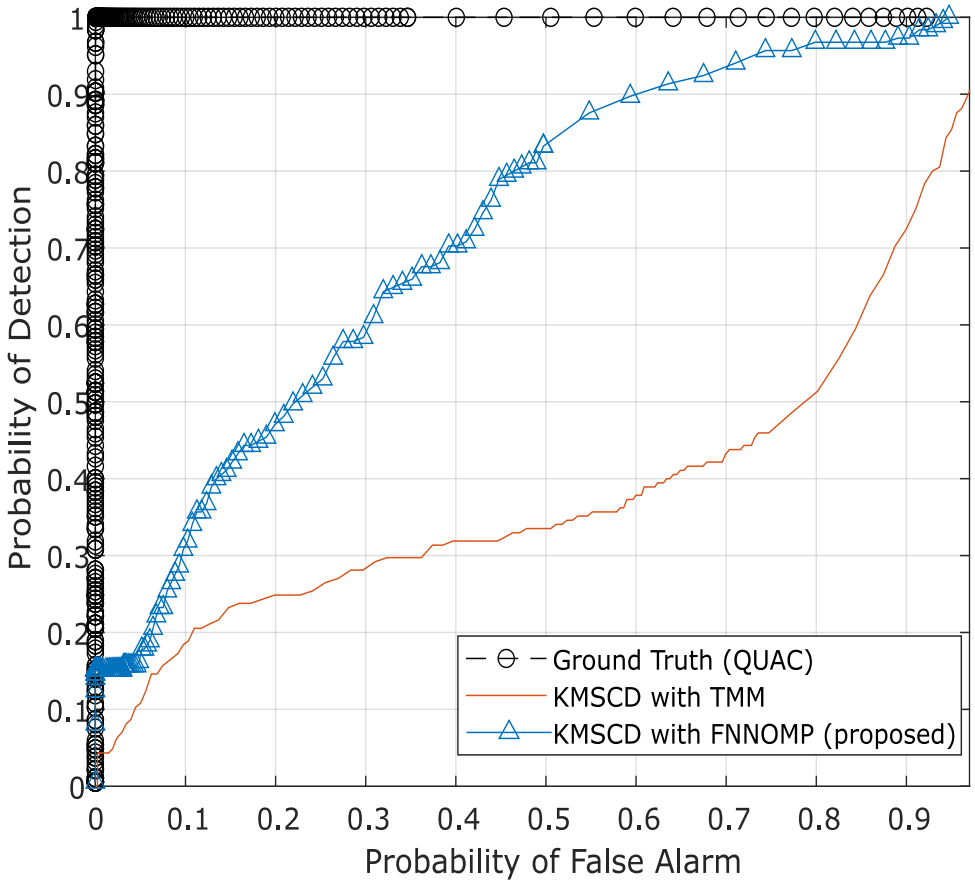


Figure 5.32: The ROC curve of Orange Perspex (OP) target material with adaptive cosine estimator (ACE). CE detector showing a better recovery of the trace material (Orange Perspex) from the one reconstructed by the Algorithm 12 with an area under the curve (AUC) of 0.68, which is almost twice as that constrained by TMM (AUC = 0.37). [3]

5.4 Rapid OMP: a proposed approach to Orthogonal Matching Pursuit (OMP)

In addition to the dictionary learning approach, a small contribution is made in sparse representation, namely in the popular orthogonal matching pursuit (OMP) algorithm. The rapid OMP proposes to iterate over *per-atom case* and achieves over fifty times improvement in computational time according to the synthetic experiment conducted while keeping the OMP algorithm's procedure intact. Since the r-OMP's procedure is the same as the OMP covered in chapter 3, the speed improvements seen is in the efficiency in matrix computations and the inefficiency in a native for-loop with modern languages like MATLAB.

OMP algorithms are solved by iterating for each signal sample, in two substeps. In scenes like AVIRIS images, the number of samples (S) can be in millions, thus, increasing the computational time drastically by dictionary learning algorithms using OMP for representation. Realising that the number of signal samples is far greater than the number of dictionary atoms, the proposed rapid OMP (r-OMP) iterates over the number of atoms, approaching atom selection and residue update the following way:

The first step is the atom selection step, which is to find out which atom has the most correlation with each signal. While the absolute value of $D^T y$ (say ϕ) is a matrix operation, the selection criteria for the maximum value require a different approach in order to avoid a loop for all samples. For each atom in a loop with counter k ($1 \leq k \leq$N), r-OMP finds if it has the maximum value for all the samples by

$$V_{k,y} = \begin{cases} 1 & \text{when } \sum_j^N (\phi_j - \phi_k > 0) = 0 \\ 0 & \text{when } \sum_j^N (\phi_j - \phi_k > 0) \neq 0 \end{cases} \tag{5.2}$$

The samples for which kth atom is with the highest residue will have no positive value in any 'j' in the expression $\sum_j^N (\phi_j - \phi_k)$ and $V_{k,y}$ will return a binary Boolean value

for kth atom and signal sample y. The constituents of V are

$$V_{k,y} = \begin{cases} 1 & \text{atom selected for representation} \\ 0 & \text{atom not selected for representation} \end{cases} \tag{5.3}$$

The second step of OMP is to estimate the representation for the selected atoms. The inverse of orthonormal with ℓ_2 norm basis is a simple transpose operation (lemma 5.4.1) such as a complete dictionary with PCA basis.

Lemma 5.4.1. *The inverse of an orthogonal and ℓ_2 norm atomic dictionary (D) is the transpose of D*

Proof.

$$\text{Let } D^{-1} \text{ be } D^T + \phi \tag{5.4a}$$

$$\implies I = D^{-1}D = (D^T + \phi)D$$

$$\implies D^T D + \phi D = I, \text{ or, } \phi D = I - D^T D$$

$$\implies \phi = (I - D^T D)D^{-1} \tag{5.4b}$$

substituting D^{-1} assumption back into the equation,

$$\phi = (I - D^T D)(D^T + \phi)$$

$$\implies \phi = (I - D^T D)D^T + (I - D^T D)\phi$$

$$\implies I\phi - (I - D^T D)\phi = (I - D^T D)D^T$$

$$\implies (I - I + D^T D)\phi = (I - D^T D)D^T$$

$$\implies D^T D\phi = (I - D^T D)D^T$$

$$\implies \phi = (D^T D)^{-1}(I - D^T D)D^T \tag{5.4c}$$

With the above ϕ, inverse of D is,

$$D^{-1} = D^T + (D^T D)^{-1}(I - D^T D)D^T \tag{5.4d}$$

In the above equation (5.4), $D^T D$ is a symmetric matrix with row/column is equal to the number of atoms in dictionary, and the exact recovery of the inverse of D is

dependant on $D^T D$ matrix being positive definite. For a dictionary with ℓ_2 orthonormal atoms, $D^T D = I$ and

$$D^{-1} = D^T \tag{5.5}$$

□

The representation update stops at this stage for orthonormal atoms, however, the basis for overcomplete sparse representations is rarely orthonormal. The inverse is estimated by least squares function in equation (5.6) where D_y refers to the dictionary subset that is selected by OMP and y is the signal sample.

$$\text{representation } a = \underbrace{(D_y^T D_y)^{-1}}_{\alpha} \underbrace{D_y^T y}_{\beta} \tag{5.6}$$

There are multiple possibilities to implement the solution to incorporate V on different programming languages. Although the estimation of β is highly parallel and it is performed in a single step, however, the estimation of α in equation (5.6) requires an additional loop in each selection of atoms in $V_{k,y}$. This can be estimated easily through existing methods in common programming languages like in MATLAB [93] and through the NumPy package in Python [94]. The complexity of the algorithm increases with increase in the combination possibilities of material mixtures, but, in increase in computational complexity per iteration is justified in real hyperspectral images where the number of signal samples are often >1000x more than the number of dictionary atoms. The algorithm estimates the representation for OMP by this approach, and estimating the residue by

$$\text{residue } r_y = y - D\alpha_y\beta_y \tag{5.7}$$

The mini-code for rapid OMP is written down in algorithm 13.

Algorithm 13 Proposed rapid OMP (r-OMP)

Input signal samples/pixel 'Y', dictionary 'D', maximum number of atoms per sample/pixel 'm', and 'n' noise/error tolerance.
Let representation 'a' and binary matrix V be two zeros matrix with dimensions of number of atoms (N) by number of samples/pixels (S).
Initialise with $r_s = Y$ and Y_s be all pixels in the image.
for j = 1 to m **do**
    $\phi = |D^T r_s|$
    for k = 1 to N **do** ▷ for all atoms
        Estimate V_k from equation (5.2)
    $\beta_e = D^T Y_s$ ▷ from equation (5.6)
    $W_s, p_s \leftarrow$ unique(V) ▷ unique columns W_s and position/index p_s
    Let w be the total number of elements in W_s
    for e = 1 to w **do**
        $D_e \subset D$
        $a_e = (D_e^T D_e)^{-1} \beta_e$
    $a \leftarrow a \bigcup a_e$ ▷ Update representation values for 'e' positions
    r = Y - Da ▷ equation (5.7)
    $Y_s \subset Y$ ▷ subset samples whose $\sum |r_y| > n$
    $r_s \subset r$ ▷ subset to residue of Y_s
Output representation 'a'

5.4.1 Experimental trials

An experiment is performed to test the improvements of r-OMP. Results presented are run on MATLAB R2019b on a Windows 10 version 1909 PC with Intel i7-6700K 4GHz processor and 32GB of RAM. The result presents a synthetic run with 10,000 (or 10k), 50k, and 100k samples with 30 dictionary atoms at 3 dictionary atoms per sample, presented with OMP, f-OMP, g-OMP and the proposed r-OMP. The experiment in figure 5.33 shows speed advances with the proposed r-OMP, completing in ≈15%, 7%, and 1.3% of the time taken by f-OMP, g-OMP, and OMP respectively.

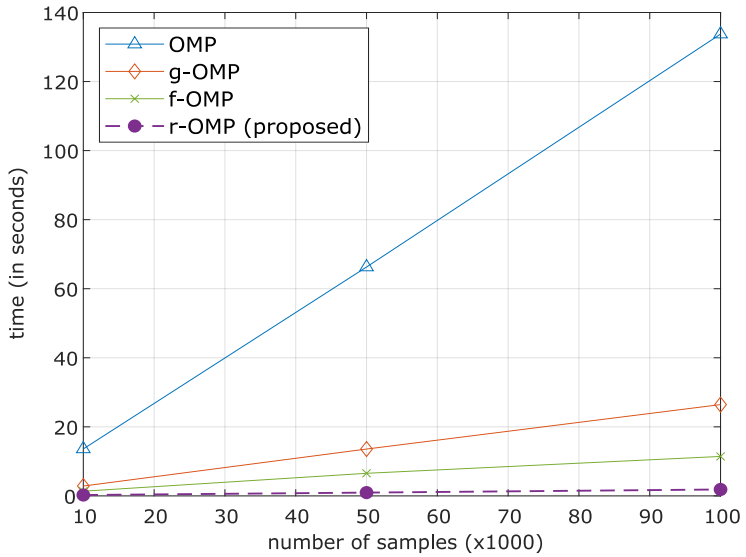


Figure 5.33: Time taken by variants of OMP

While synthetic experiments provide some conclusion, it is crucial to present the impact on a real scene. An experiment presented with the popular dictionary learning algorithm, K-SVD [13], which uses OMP for representation, is tested on the Selene H23 Dual scene. K-SVD is run on the H23Dual image with 10, 20, and 30 atoms for a maximum of three atoms per pixel, and 50 training iterations. Results presented in table 5.5 show that r-OMP completed in ≈half hour each compared to OMP in ≈13.5 hours for the same mean error of ≈3 in Manhattan distance on an average. The proposed r-OMP has the potential for further acceleration by the use of parallel processing and GPU.

Table 5.5: Time taken (in minutes) by K-SVD dictionary learning on Selene H23 Dual scene for 50 iterations

K-SVD with	**OMP (existing)**	**r-OMP (proposed)**
10 atoms	811.19	27.53
20 atoms	824.33	31.31
30 atoms	811.89	39.15
Average time taken	815.80 ± 7.39	32.66 ± 5.93

5.5 Chapter summary

This chapter presents two proposed dictionary learning approaches that replaces random sampling in the classic sparse coding dictionary (C-SCD) model: with

simultaneous orthogonal matching pursuit (SOMP) residue called SCD-SOMP, and KMeans clustering before sampling called KMSCD algorithm. The SCD-SOMP algorithm selects samples with the maximum SOMP residue in attempt to minimise the infinite norm. With SCD-SOMP the benefit is that no endmember (EM) material (be it background and trace) is left with a high residual error when the model converges, and the drawback with minimising the infinite norm is a higher mean error relative to other models in its class.

The second approach presented the KMSCD which utilizes pre-classification of the MSI scene to extract material species of the imagery. The method uses the popular KMeans algorithm for unsupervised classification. With a pre-classification and selection of 1 sample per class, the probability to learn the different/unique materials in the scene is increased drastically and the model receives consistent samples for training in each iteration. This method increases the ability to include small (trace) targets in the reconstructed scene, and it also reduces reconstruction error in comparison to other unmixing methods. KMSCD is further presented as a feasible material allocation method by the incorporation with a fast non-negative orthogonal matching pursuit (FNNOMP) to estimate the abundance (representation) of the mixed pixels in the scene. FNNOMP helps constrain the EMs to the maximum four EM per pixel to fulfil the limitation required by the CameoSim simulator.

An additional work is done to speed up the popular orthogonal matching pursuit (OMP) using a modified version which we name it as rapid OMP (r-OMP). Modern computer languages are more optimised for matrix operations compared to using native for loops. The proposed rapid OMP (r-OMP) simultaneously estimates all samples by the elimination of for loop in the OMP. Experiment on a real scene with a popular dictionary learning algorithm, K-SVD, have shown that the proposed r-OMP completes K-SVD in $\approx$4% of the computational time compared to using OMP.

Chapter 6

Scene Simulation

In the previous chapter, the classic sparse coding dictionary (C-SCD) learning algorithm with random sampling was replaced with k-means clustering for sub-sampling selecting one pixel from each cluster in the proposed method, called k-means sparse coding dictionary (KMSCD) unmixing coupled with fast non-negative orthogonal matching pursuit (FNNOMP) for the abundance optimised the endmember (EM) materials for external classification in CameoSim's material allocation module. The KMSCD+FNNOMP reconstructing the reflectance HSI from all band input produced $\approx$10x improvement (and $<$1% mean DL1NE) compared to using the existing texture material mapper (TMM) for abundance.

The purpose of a scene simulator is in its ability to simulate the scene faithfully. This chapter first provides a brief background in atmospheric compensation in section 6.1, followed by a step by step explanation in the scene simulation process, and the proposed material allocation module used externally from CameoSim simulation. Scene simulation is performed with proposed methods compared with using the TMM for abundance in section 6.2 and the accuracy of simulated scene which includes the accuracy of all processes combined together: atmospheric compensation, material allocation, and scene simulation with approximated atmosphere. Towards the end of this chapter, an additional work is presented which is published apart from the material allocation work. This work proposes a novel method to approximate in-scene white and black calibration panels in section 6.3 which helps in detection of shadows.

6.1 Atmospheric Compensation

An HSI scene simulator uses radiative transfer (RT) equations to mimic how light interacts with real world objects, converting reflectance of a material to radiance for the given atmosphere. The ultimate goal of an atmospheric compensation algorithm is to reverse the process and convert a given radiance to reflectance, which is to remove the effects of atmospheric absorption and scattering, as well as illumination angle artefacts by converting the at-sensor radiance to the reflectance of the target surface.

There are three types of atmospheric compensation methods: empirical methods, model based methods, and hybrid methods that use a model derived from the RT equation together with an empirical approach. The first is the in-scene empirical approach like the empirical line method (ELM, [95, 96]). The Empirical Line Method (ELM) is a method that approximates the target reflectance by field measurements without considering possible effects such as the topography of the scene. The ELM equation is written in (6.1) for a given wavelength 'λ' on target 't', where ρ is the reflectance of a pixel, L is the radiance, and L_w and L_b are radiance data for white and black pixels of the scene.

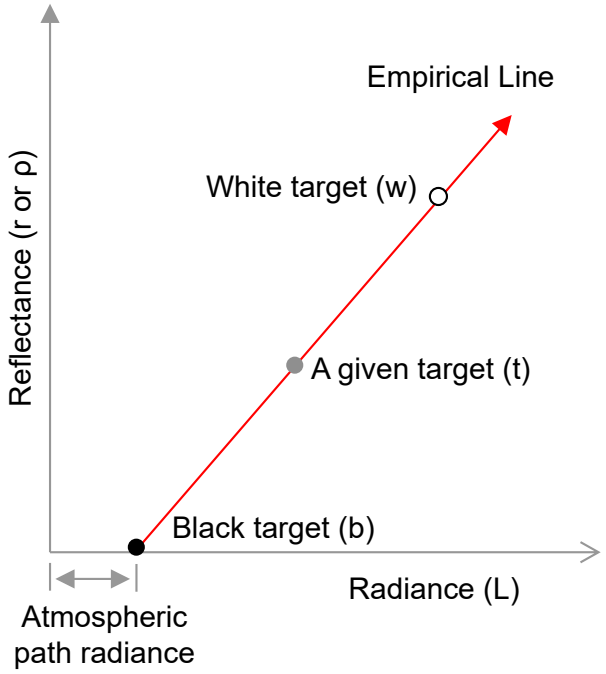


Figure 6.1: An illustrative example of an empirical line (the image is modified from [96])

$$\rho_t(\lambda) = \frac{L_t(\lambda) - L_b(\lambda)}{L_w(\lambda) - L_b(\lambda)} \tag{6.1}$$

The second type of atmospheric compensation methods are model based like the Atmosphere Removal Algorithm (ATREM) which is often used to compensate AVIRIS images, and Quick Atmospheric Correction (QUAC, [72]). Due to the proprietary nature of the algorithm, the exact implementation is unknown, however, according to the publication in [72], the flowchart of the algorithm is given in figure 6.2. QUAC is an advanced atmospheric compensation method, that only requires approximate central wavelengths of each spectral band. QUAC works without requiring *a priori* information of the atmospheric conditions of a given scene, and with the assumption that a given scene has at least 10 diverse materials, which is suitable for a scene like Selene.

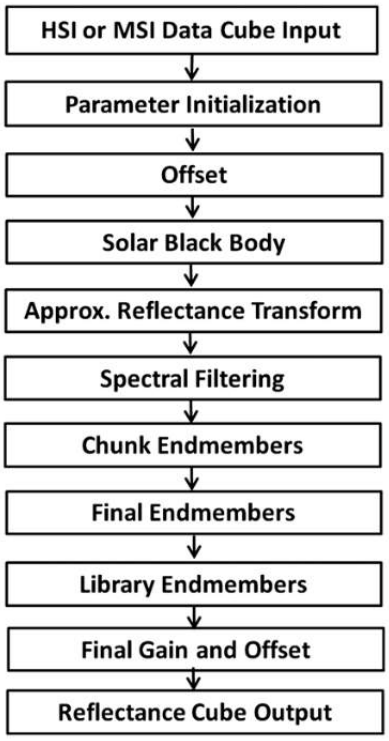


Figure 6.2: Flow diagram of QUAC atmospheric compensation, refer [72]

Algorithms like the Atmospheric and Topographic Correction (ATCOR) and the Fast Line-of-Sight Atmospheric Analysis of Spectral Hypercubes (FLAASH) [18] require detailed atmospheric information of the scene for compensation, thus, not feasible for the

Selene scene. In search to approximate the simulating atmospheric conditions for the Selene scene, the paper in [24] is proposed, which approximates white and black targets of a given region in scene using a combination of radiance, QUAC reflectance, and the ELM property. A by-product of the proposed method can be utilised for classification of shaded regions.

6.2 Scene simulation with proposed material allocation

The main objective of this research is to reproduce the reflectance of ground truth data with minimal error within a simulator from material allocation to scene simulation. To recall the previous chapters, the C-SCD dictionary learning model was used for unmixing, extracting EMs solving the optimisation problem. Further, the sub-optimal random sampling in C-SCD was replaced with k-means clustering selecting 1 pixel from each cluster, called k-means sparse coding dictionary (KMSCD). Finally, the KMSCD is merged with fast non-negative orthogonal matching pursuit (FNNOMP) algorithm to accommodate for the four material per pixel limit in CameoSim's material allocation module, which achieved $\approx$10x improvement compared with the existing TMM.

With the existing and proposed material allocation models, a scene simulation with theoretical approximation of the atmosphere will be better suggestive of the proposed improvements. This section presents the experimental setup to simulate the Selene scene from start to finish in section 6.2.1, and the results of scene simulation showing the benefits of using the proposed approach in section 6.2.2.

6.2.1 Experimental Setup

Scene simulation is performed in CameoSim using texture tools, geometry tools, project and database tools, and imagery tools module. This section explains the step by step procedure to simulate a scene, providing explanations of each module. The first step is to decompose the potential endmembers (EMs) using an hyperspectral input with the proposed KMSCD+FNNOMP method with the four material per pixel constraint. With the generated EMs, a material allocation map is created in a 'CTX' file format as used in

CameoSim (figure 6.3 showing the contents in the file). Each EM material is its own file storing 'wavelength' and 'reflectance' pair in a tabular format. The MATLAB implementation of the proposed material allocation module with KMSCD+FNNOMP bypassing CameoSim's internal methods is available in appendix C.2, and the source code to prepare CameoSim's CTX file is presented in appendix C.3.

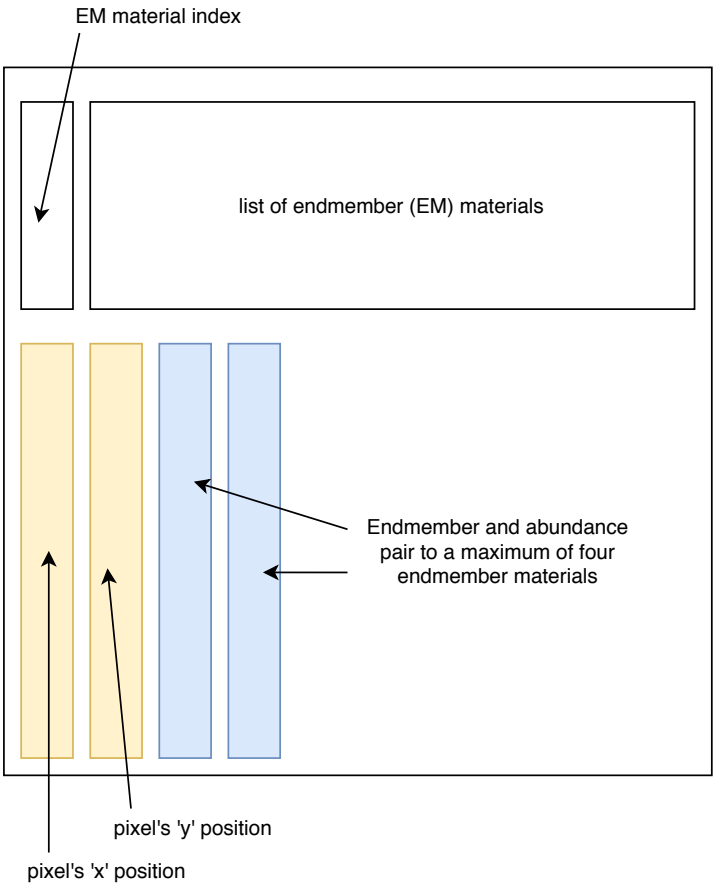


Figure 6.3: The contents in a CameoSim's CTX file for material allocation showing first a list of endmember (EM) materials, and a table with four or more columns where the first two are the 'x' and 'y' position of a pixel followed by EM index and its abundance value.

A screenshot of the CTX file created by used the bespoke material allocation method with the proposed KMSCD+FNNOMOP method is shown in figure 6.4. Once the CTX file is created bypassing the material mapper methods in CameoSim texture tools like the TMM, the second step is to create a 3D geometry and integrate it with the generated CTX in the texture tools module. A two dimensional plane is created in 3D space to generate a 2D flat surface. Figure 6.5 shows the generated CTX with flat geometry for Selene scene.

```
# CameoSim image classification data

I /opt/insys/share/cameosim/materials/KMSCDFNNOMP_SeleneH23Dual_E20_WV2_image.png

M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_1_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_2_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_3_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_4_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_5_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_6_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_7_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_8_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_9_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_10_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_11_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_12_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_13_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_14_mat
M   KMSCDFNNOMP_SeleneH23Dual_E20_WV2_endmember_15_mat

N 4
C   0   0   0  51   1  48   2   1
C   1   0   1  49   0  48   3   3
C   2   0   1  75   0  18   3   7
C   3   0   1  75   0  18   3   7
C   4   0   1  75   0  18   3   7
C   5   0   1  75   0  18   3   7
C   6   0   1  76   0  18   3   6
C   7   0   1  76   0  18   3   6
C   8   0   1  76   0  18   3   6
C   9   0   1  76   0  18   3   6
C   10  0   1  76   0  18   3   6
C   11  0   1  77   0  18   3   5
C   12  0   1  76   0  18   3   6
C   13  0   1  76   0  18   3   6
C   14  0   1  76   0  18   3   6
C   15  0   1  77   0  18   3   5
C   16  0   1  77   0  18   3   5
C   17  0   1  76   0  18   3   6
C   18  0   1  76   0  18   3   6
C   19  0   1  76   0  18   3   6
C   20  0   1  76   0  18   3   6
C   21  0   1  75   0  18   3   7
C   22  0   1  50   0  47   3   3
C   23  0   1  76   0  18   3   6
```

Figure 6.4: Contents of an example CTX file created using the bespoke external material allocation module for CameoSim simulator. The CTX file shown is material allocation from the centre wavelengths of WorldView-2 (WV2) as input in the test image. The content is structured according to the required input, shown previously in figure 6.3.

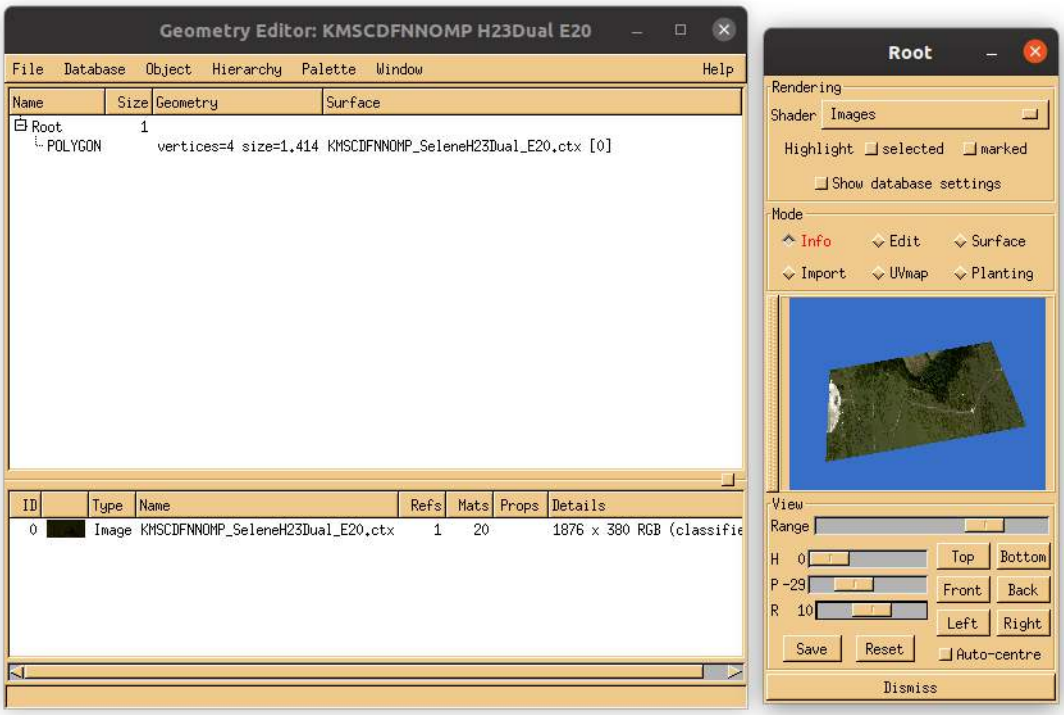


Figure 6.5: Texture tools in CameoSim showing a polygon with four vertices for a rectangular 2D plane in 3D space, and the CTX file generated with the proposed material allocation method.

The next step is to create a new project in the database tools and define the parameters of simulation. Scene simulation results presented in this thesis is done at the highest possible quality, on a maximum of eight CPU threads, taking an average of ≈5 hours per simulation. Regular grid sampling scene is used where a single pixel is sub-divided into 'n^2' (where 'n' is an integer) number of sub-subsections with 'n' rows and 'n' columns, and a single ray is fired on each sub-section. Regular grid sampling with 100 rows and 100 columns is used for simulation, for the high accuracy firing 10k rays to each pixel. The sensor is placed on a height of 944.8m from the ground (2D flat plane) looking directly at the plane at nadir. And, a systematic study by researchers in this DSTL project converged to the following double layer cloud for the Selene scene:

Table 6.1: Double layer cloud parameters approximated for Selene scene

Cloud	**Layer 1**	**Layer 2**
Base height (in km)	0.6	1.2
Thickness (in km)	0.01	0.1
Water droplet density (in g/m^3)	0.01	0.1
Extinction coefficient (per km)	1.875	18.75

The scene is simulated with the assigned project properties in database tools and material allocation combined with geometry in texture tools. The final step after simulation is to export the image so it can be read outside of the CameoSim. Imagery tools module contains a function that with it which is used to export a simulated image to Flexible Image Transport System (FITS) file format. FITS is an open standard for digital image storage, which can be read in external programs like MATLAB and Python. The inputs for the export (shown in figure 6.6) are (i) the imagery source which is the simulated/processed project file, (ii) start and finish frames of the project where a single frame is used for still imagery and more than one frame is for a hyperspectral movie file, and (iii) the output file name.

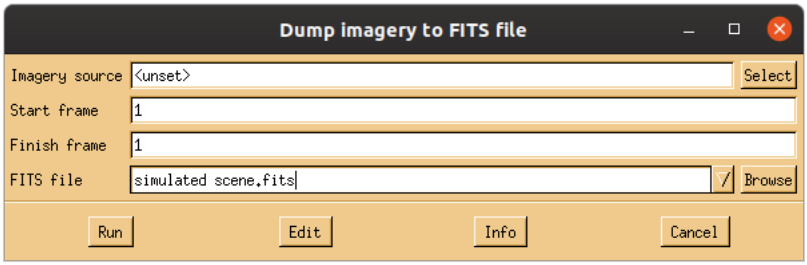


Figure 6.6: CameoSim imagery tools module to export a simulated scene to FITS file format, taking inputs like the source which is simulation project name, simulated frames, and export file name.

The flow diagram of the complete process from material allocation with the proposed method to exporting the simulated imagery is shown in figure 6.7.

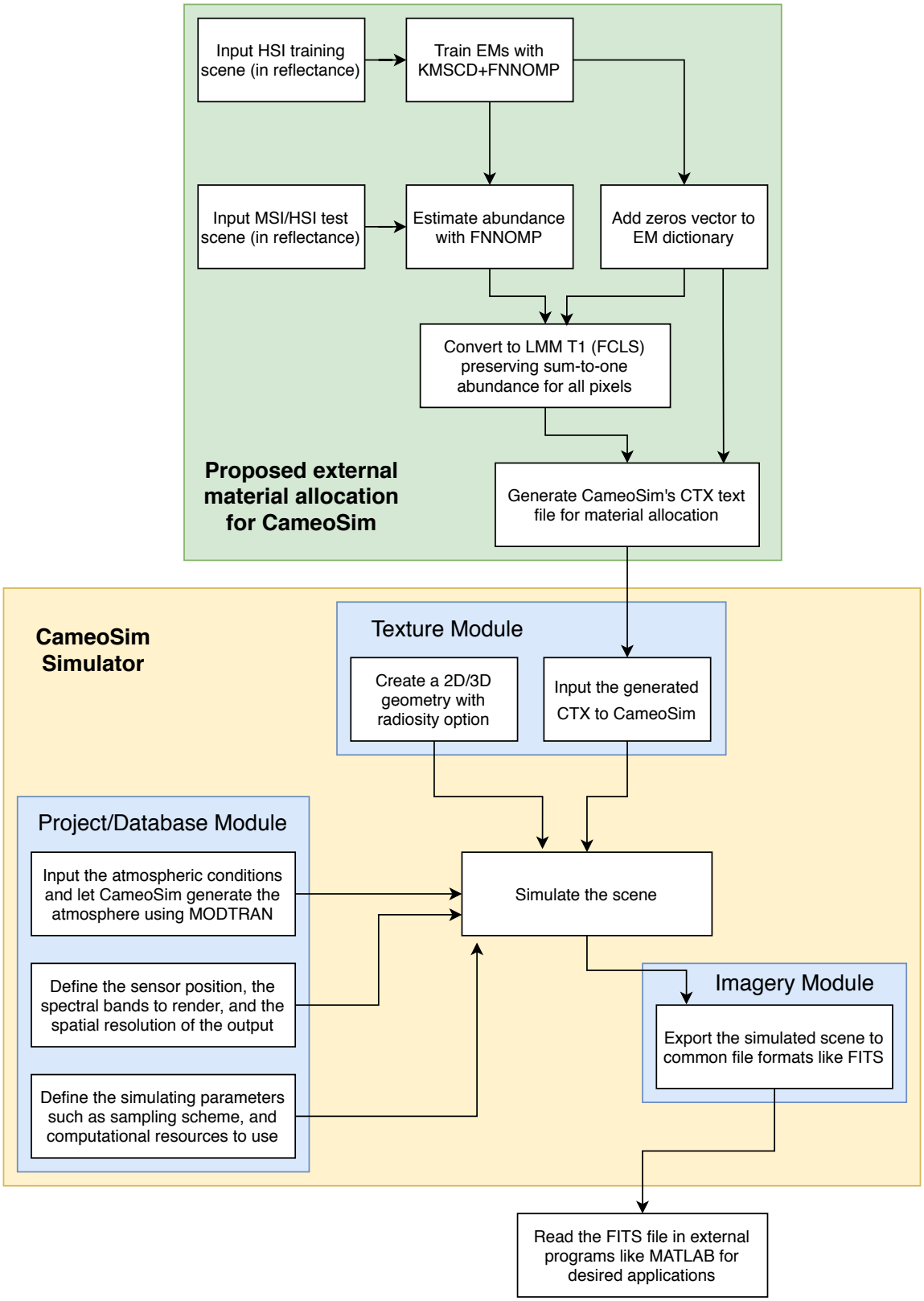


Figure 6.7: Flow diagram from MSI/HSI input image to exporting the simulated scene, replacing CameoSim's in-built material mapper methods with the proposed approach to material allocation.

6.2.2 Simulation Results

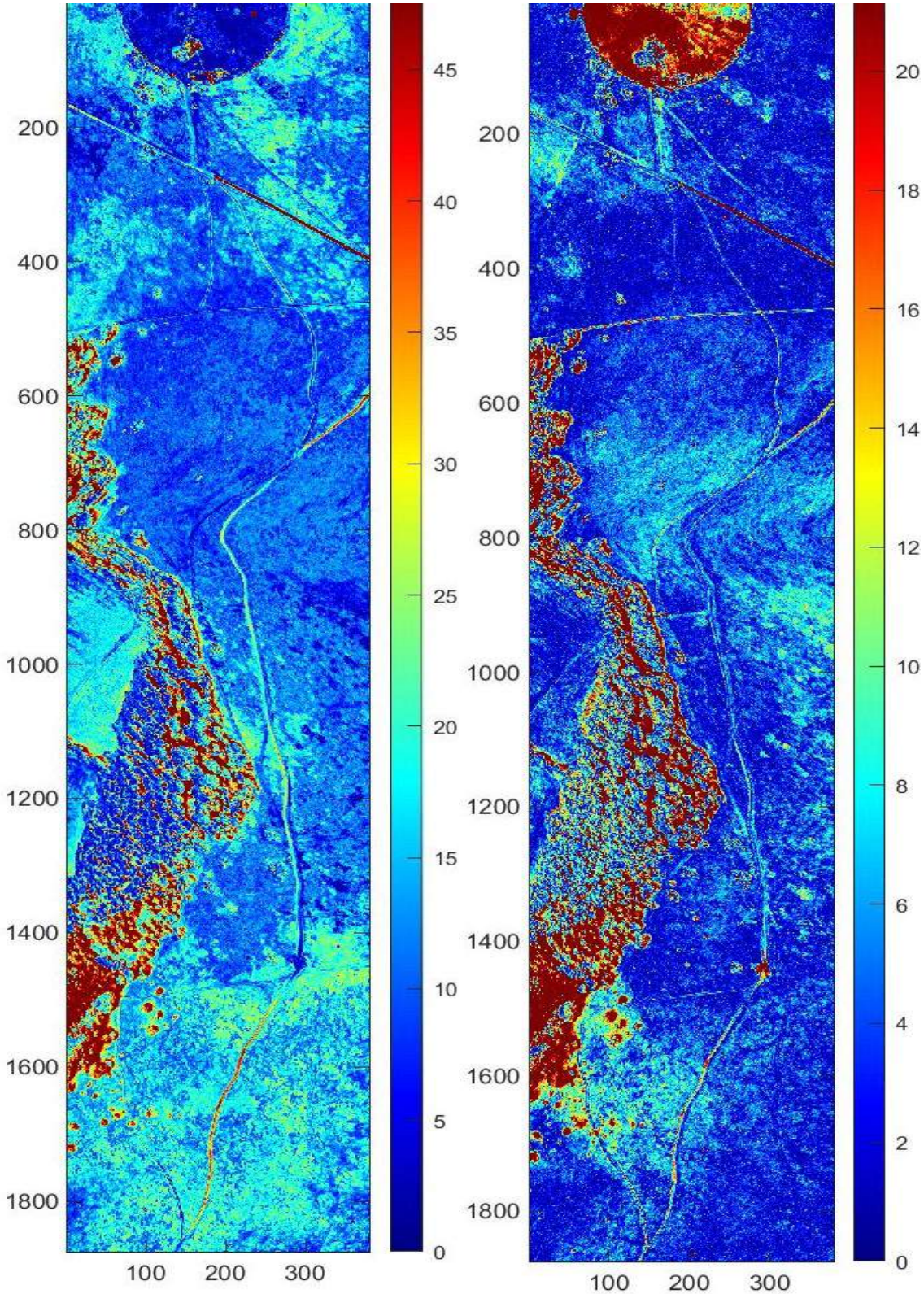
This section presents the simulated result of the proposed external material allocation module to CameoSim, as presented in figure 6.7, using the optimised parameters of atmosphere with the double layer cloud. The estimation of the material abundance of the Selene scene by using TMM has the mean DL1NE error 15.97% with respected to the ground truth. The same experiment by using the proposed KMSCD+FNNOMP method gives mean DL1NE error of 7.07%, i.e., the TMM method is ≈2.26x higher error than that by using the proposed FNNOMP method. This error is inclusive all intermediate errors of the sub-steps: atmospheric compensation with QUAC, material allocation, approximation of atmosphere, and errors from CameoSim's rendering. Furthermore, with MSI input for reconstruction such as the 8 centre wavelengths of the WorldView-2 (WV2) and WorldView-3 (WV3), the proposed method using FNNOMP has shown an 86% (or 2.45x) overall improvement in the mean DL1NE and 30% improvement in the Manhattan distance (MD) error compared to using TMM for abundance. Reconstruction from MSI input use the HSI for training. Tables 6.2 and 6.3 summarises the DL1NE and MD errors, with DL1NE error map in figures 6.8 for all band input, 6.9 for centre wavelengths of WV3 as input, and 6.10 for the centre wavelengths of WV2 as input. The combined results show that the proposed approach to material allocation and external module improves CameoSim's scene simulation accuracy by >2 times.

Table 6.2: Mean DL1NE radiance error for the simulated scenes

KMSCD with	**FNNOMP** (proposed)	**TMM**
All band input	7.07	15.97
WV3 input	7.23	19.92
WV2 input	12.84	30.67
Mean	**9.04**	**22.19**

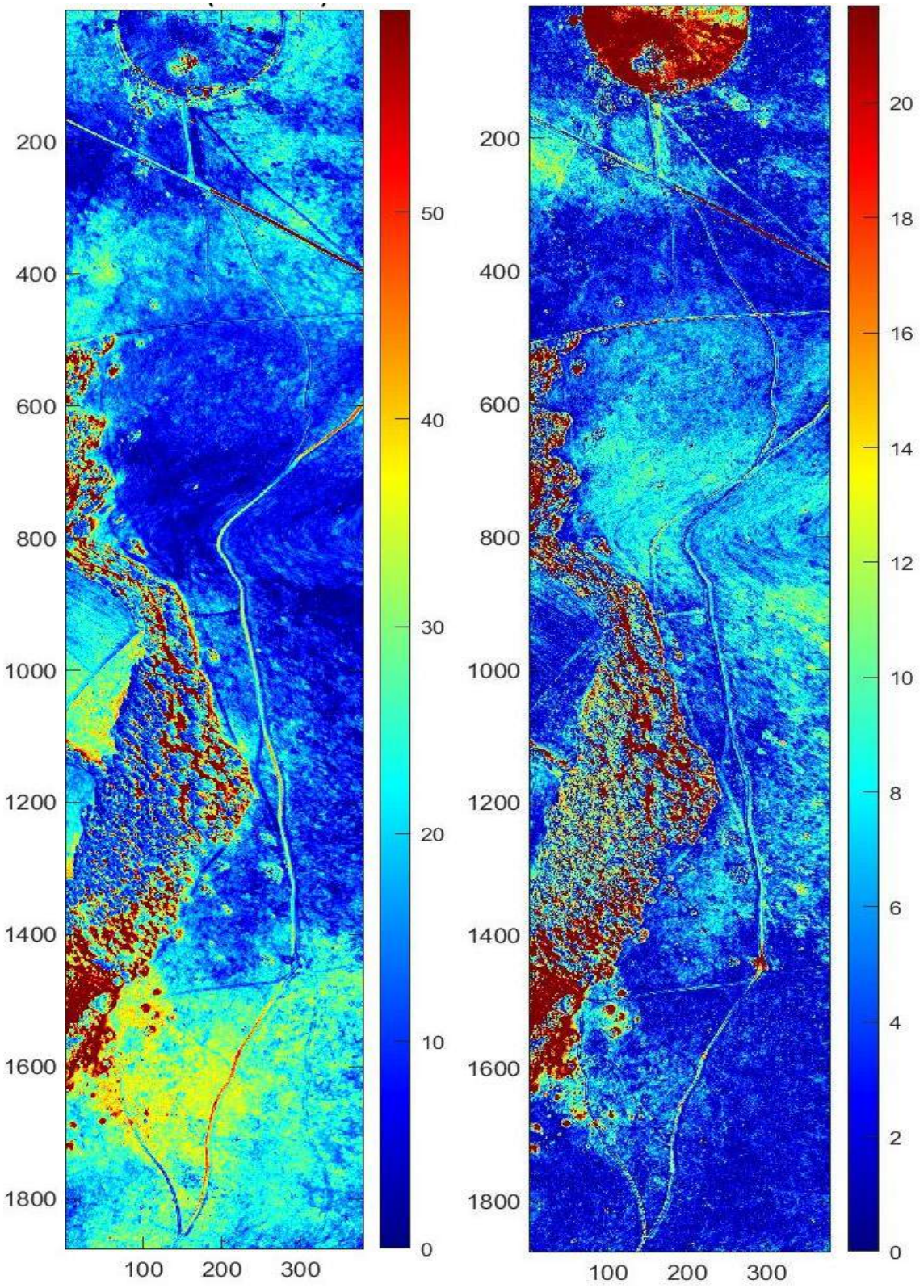
Table 6.3: Mean MD radiance error for the simulated scenes

KMSCD with	**FNNOMP** (proposed)	**TMM**
All band input	1.04	1.35
WV3 input	1.06	1.70
WV2 input	1.33	1.81
Mean	**1.14**	**1.62**



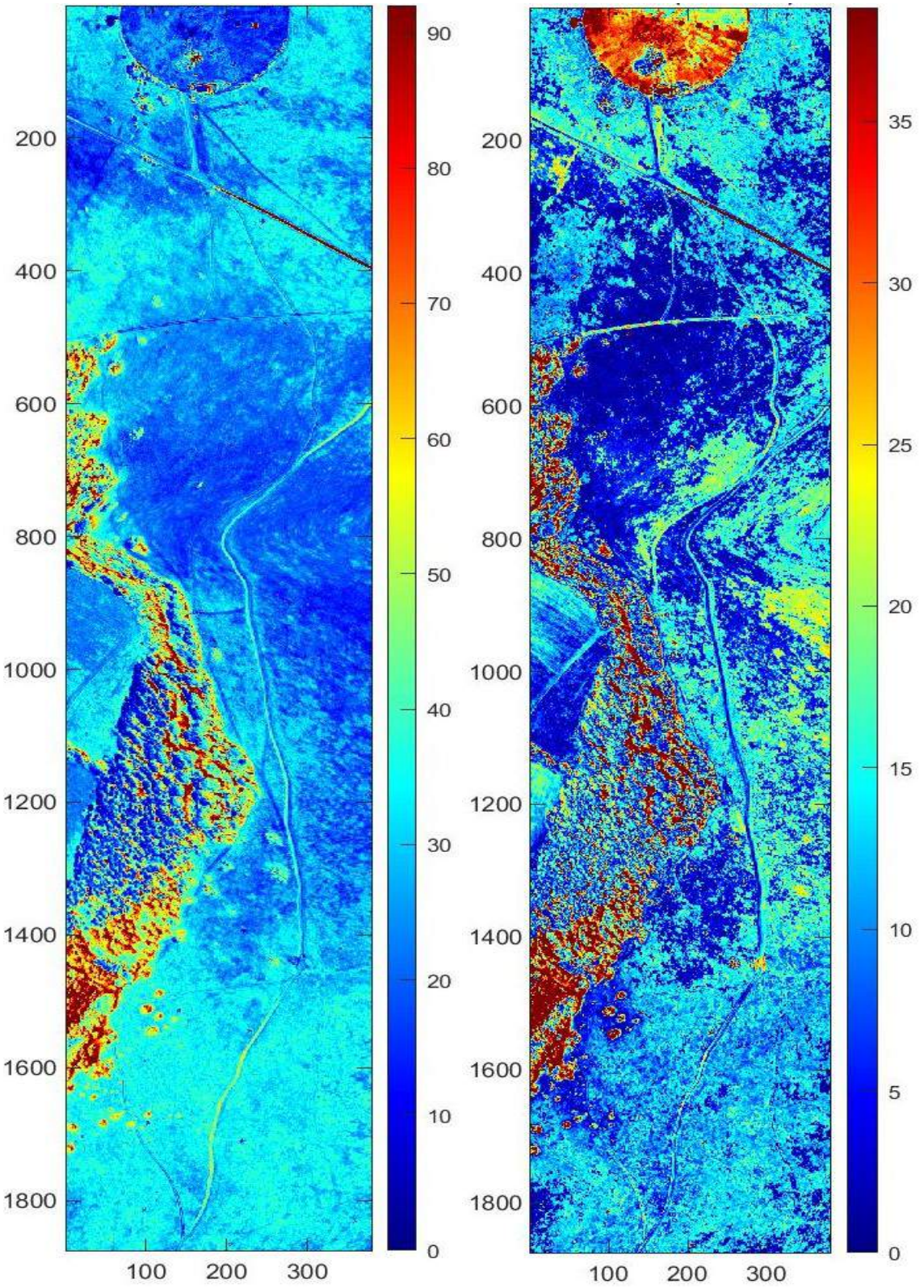
(a) KMSCD+TMM (b) KMSCD+FNNOMP

Figure 6.8: DL1NE radiance error of the simulated scene with the proposed KMSCD unmixing with all band input, where abundance is estimated with (a) TMM with a mean error of 15.97%, and (b) FNNOMP with a mean error of 7.07%.



(a) KMSCD+TMM (b) KMSCD+FNNOMP

Figure 6.9: DL1NE radiance error of the simulated scene with the proposed KMSCD unmixing with the centre wavelengths of WV3 as input, where the abundance is estimated with (a) TMM with a mean error of 19.92%, and (b) FNNOMP with a mean error of 7.23%.



(a) KMSCD+TMM (b) KMSCD+FNNOMP

Figure 6.10: DL1NE radiance error of the simulated scene with the proposed KMSCD unmixing with the centre wavelengths of WV2 as input, where the abundance is estimated with (a) TMM with a mean error of 30.67%, and (b) FNNOMP with a mean error of 12.84%.

6.3 Proposed approach to white and black panel approximation and detection of shaded regions

An additional work is done in this thesis apart from the material allocation work for CameoSim simulator, which is an approach to blindly approximate white and black panel localised to a scene which is further used to estimate a shadow map. The proposed technique approximates white and black calibration panels of a localised area through empirical method. In principle, a ray of light shines from the sources, bounces off a surface with material reflective properties, and arrives at the sensor. A black panel (reflectance = 0) captures all the incoming light off the surface and atmospheric path radiance arrives at the sensor. And, a white panel (reflectance = 1) reflects off the total intensity of light. Spectral knowledge of black and white panel provides detailed information of the upwelling and downwelling radiance of the scene which can be used together to estimate the solar irradiance of the scene.

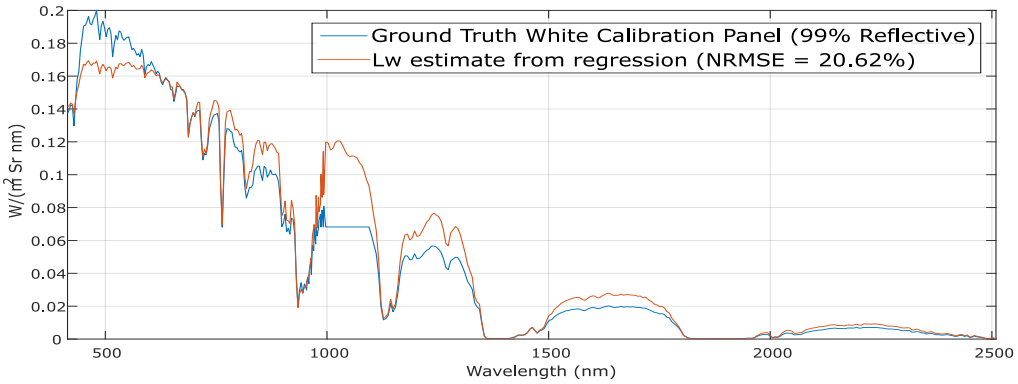
Approximation of the panels is useful in scenes that does not have them in-scene. The proposed method achieves this by taking into account the at-sensor radiance and its reflectance as inputs, where reflectance estimated by a model-based atmospheric compensation method like QUAC which does not require any prior information of the atmospheric conditions of the scene. Given radiance and reflectance of two or more materials, the slope and intercept of the empirical line can be deduced, and thus approximating white and black panel when reflectance 'r' is 1 and 0 respectively. The mathematical rationale for blind approximation of the calibration panels for radiance 'L', radiance of white panel 'L_w', radiance of black panel 'L_b' and reflectance 'r' is:

$$\text{By ELM equation, the reflectance r}(\lambda) = \frac{L(\lambda) - L_b(\lambda)}{L_w(\lambda) - L_b(\lambda)} \tag{6.2a}$$

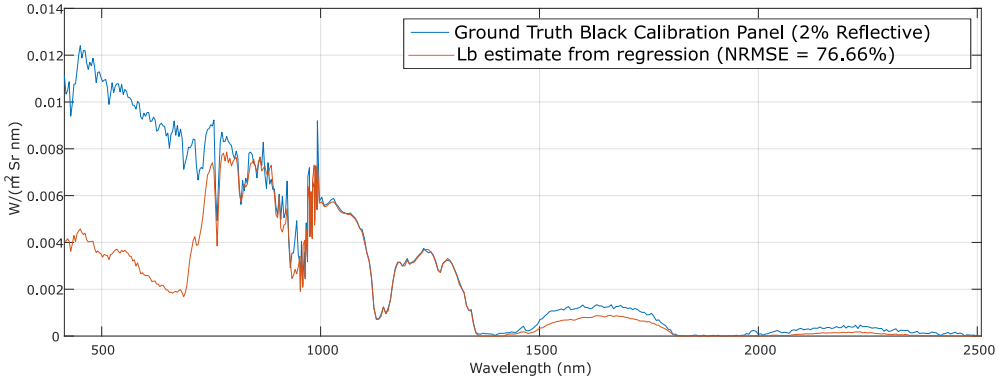
$$\implies (L_w(\lambda) - L_b(\lambda))r(\lambda) = L(\lambda) - L_b(\lambda)$$

$$\implies L(\lambda) = \overbrace{(L_w(\lambda) - L_b(\lambda))}^{\text{slope}}\, r(\lambda) + \overbrace{L_b(\lambda)}^{\text{intercept}} \tag{6.2b}$$

The equation (6.2) approximates the white and black calibration panels given at-sensor radiance and reflectance for a localised area consisting of two or more different materials by use of linear regression. The white and black panel estimates are shown in figure 6.11.



(a) White Panel Estimation



(b) Black Panel Estimation

Figure 6.11: White and black panel estimates from global search with the normalised root mean square error (NRMSE) of 20.62% and 76.66% for white and black panel respectively [24].

With the above methodology, the approximated white and black panels for the whole scene has resulted in 20.62% and 76.66% normalised root mean square error (NRMSE) compared to ground truth calibration panels. The magnitude of these errors appears to be significantly large but it is difficult to verify this error in this data set due to the lack of ground truth shadow map and detailed radiometric measurement across the entire scene. The purpose of this section is to propose the idea and the validation of this methodology remains to be part of the future work. However, the above information can

be extracted from two different localised areas: a given HSI image which is a few meter square in area, and its nearest neighbours. The estimates from the whole scene is referred to as global, and the nearest neighbours will be referred to as local estimated by a sliding window approach. With global and local parameters, the non-shaded radiance 'L_{NS}' is given by:

$$\text{Reflectance } r(\lambda) = \overbrace{\frac{L(\lambda) - Lb_{SW}(\lambda)}{Lw_{SW}(\lambda) - Lb_{SW}(\lambda)}}^{\text{Local/Sliding Window}} = \overbrace{\frac{L_{NS}(\lambda) - Lb_G(\lambda)}{Lw_G(\lambda) - Lb_G(\lambda)}}^{\text{Global/Whole Scene}} \tag{6.3a}$$

Estimating for non-shadow radiance $L_{NS}(\lambda)$,

$$L_{NS}(\lambda) = \frac{L(\lambda) - Lb_{SW}(\lambda)}{Lw_{SW}(\lambda) - Lb_{SW}(\lambda)} \times (Lw_G(\lambda) - Lb_G(\lambda)) + Lb_G(\lambda) \tag{6.3b}$$

$$\text{Or, } L_{NS}(\lambda) = (L(\lambda) - Lb_{SW}(\lambda))m(\lambda) + Lb_G(\lambda)$$

$$\text{Where, } m(\lambda) = \frac{Lw_G(\lambda) - Lb_G(\lambda)}{Lw_{SW}(\lambda) - Lb_{SW}(\lambda)} \tag{6.3c}$$

Equation (6.3) gives the non-shaded radiance from local and global estimated white and black calibration panels. The relative ratio between the at-sensor radiance and the estimated non-shaded radiance gives a relative illumination factor, which is used for shadow masking. The relative ratio extracted for Selene H23 Dual scene is shown in figure 6.12b. The shadow map in figure 6.12c has been chosen with a user-defined threshold of 0.95 and this process is not yet automatised. The threshold was chosen arbitrarily for illustration only due to the lack of ground truth of the shadows in the scene, and the appropriate threshold for the indication of the shadows will be part of the future work.

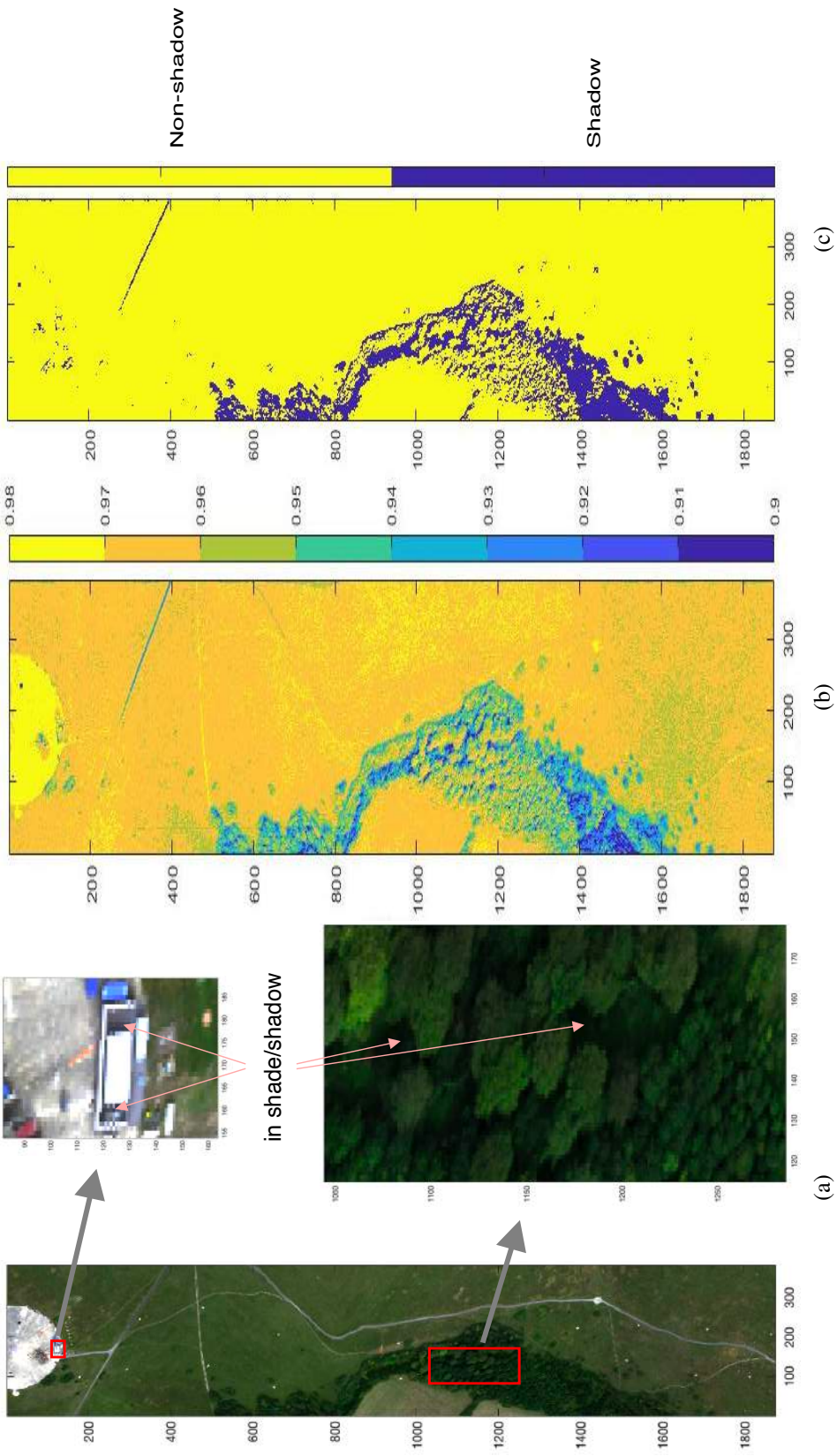


Figure 6.12: To illustrate how the proposed method may be able to use for shadow masking and possibly shadow compensation (a) shows the RGB of Selene H23 Dual image, (b) shows the relative illumination factor for Selene H23 Dual scene where the colorbar scaled between 0.9 and 0.98 for visual presentation, and (c) shown an exported binary shadow map. [24]

Chapter 7

Conclusions and Future Work

The main objective of this research was to develop a faithful scene reconstruction algorithm to be implemented within a hyperspectral scene simulator. This chapter summarises the contributions during the course of this PhD programme, and provides potential future scope which may help improve the algorithms.

Two journal publications, with SCD-SOMP in [23] and KMSCD in [3] algorithms are developed to help improve the robustness of a learned dictionary, for both the background and minority pixels. Additionally, two algorithms have been developed that complements the goals of dictionary learning: a conference submission with rapid Orthogonal Matching Pursuit (r-OMP) increasing the performance of Orthogonal Matching Pursuit (OMP) algorithm, and another journal publication in [24] blindly approximating white and black calibration panels for an unknown scene which helps in estimating shaded regions from fully illuminated ones and helps in approximating the atmosphere.

OMP algorithms traditionally estimate sparse representation by iterating over each sample individually, and an inner loop which selects one atom at a time that is most correlated with the residue. The proposed rapid OMP (r-OMP) runs the loop for 'N' atoms, simultaneously estimating for all samples, and, in a real scene since $N \ll S$, the proposed approach speeds up OMP by several orders of magnitude. A synthetic experiment shows speed advances with the proposed r-OMP, completing in $\approx$1.3%-15%

of the time taken by variants of OMP. And, the journal article in [24] provides an empirical method to approximate white and black calibration panel for a localised area given radiance and its reflectance estimated by QUAC [72], and approximate a pixel's relative illumination factor which can be used for shadow masking.

7.1 SCD-SOMP

Self-dictionary (SD) algorithms iteratively seek to find the endmember (EM) material which is least correlated with the EMs from previous iterations. Simultaneous orthogonal matching pursuit (SOMP) is one of the measures used for selection of an SD dictionary, SOMP residue estimating the residual error of dictionary atoms simultaneously. The proposed SCD-SOMP algorithm uses this pixel selection criteria of the SD algorithm to select the handful of sub-samples in C-SCD algorithm during training. The outcome of such a sample section results in the minimisation of the infinite norm or the maximum error, where neither the dominant background materials nor the trace target materials is left with a very high residual error with the learned dictionary. The achievements and limitations of the proposed SCD-SOMP are:

- In RX anomaly detection, the SCD-SOMP technique produced a correlation of 8.96e-01 in 10 atoms with the ground truth where C-SCD is at 2.81e-01. The performance in RX is found to be ≈43-100% better than C-SCD.
- The performance in ACE target detection with Orange Perspex material produced an area under the curve ≈7-20% better and ≈78-92x more consistent than C-SCD.
- The drawback of the algorithm is in minimising the mean error.

7.2 KMSCD

With C-SCD for EM extraction, the EM dictionary is learned in a sub-optimal way by random selections of pixels from the scene during a training cycle, the performance of the algorithm varies from run to run. The poor performing results are seen in detection accuracy with a correlation of 4.57e-01 (mean) with 10 to 50 EM dictionary, and 4.44e-01

(mean) on a five repeated runs with 40 EM dictionaries. Similar results were discovered in target detection with Orange Perspex material with a mean area under the curve (AUC) of 8.29e-01 with 10 to 50 EM dictionary, and 7.15e-01 on five repeated runs. The work done is the proposed KMSCD is to increase the probability of selecting different materials in the scene with k-means classification, selecting 1 pixel from each class during training. It is found that:

- KMSCD unmixing produced a correlation of 7.70e-01 (mean) with 10 to 50 atoms, which is ≈68.5% improvement compared to C-SCD unmixing. Moreover, in the reproducible test of five repeated runs with 40 EM dictionary, KMSCD produced a correlation of 7.70e-01 (mean), which is ≈75% improvement over C-SCD unmixing.

- ACE detection with Orange Perspex target material produced an AUC of 9.98e-01, which is close to perfect accuracy in 10 to 50 EM dictionary, which is ≈20.4% improvement compared to C-SCD unmixing. Moreover, in the reproducible test of five repeated runs with 40 EM dictionary, KMSCD produced an AUC of 9.98e-01, which is ≈40% improvement compared to C-SCD unmixing.

- KMSCD unmixing has 20-500% enhanced accuracy compared to state of the art unmixing methods compared in this thesis from methods with pure pixel assumptions like VCA [35] and SD-SOMP [14], methods assuming pixels are in the facets like MVSA [36], and statistical optimisation methods like CoNMF [37] and C-SCD unmixing [11, 34].

CameoSim simulator requires an EM-abundance to strictly follow an LMM with FCLS abundance for all pixels in the scene, with the added constraint of a maximum of four materials in a pixel in the current structure. A new method for material allocation is proposed with KMSCD unmixing together with the modern FNNOMP for abundance estimation, and a bespoke external material allocation module is written that works with CameoSim. The method iteratively optimises to the EM dictionary with the four material per pixel constraint required. Material allocation with the proposed method produced a mean DL1NE of ≈0.7% (<1% error), which is a 10x improvement than using TMM for

abundance. Scene simulation with the HSI input with the existing approaches in CameoSim like the TMM produced ≈2.26x the mean error compared to the proposed with FNNOMP.

7.3 Future Work

This thesis presents the development of a spectral reconstruction method for multispectral (MSI) and hyperspectral (HSI) applications through an enhanced dictionary learning and spectral unmixing methodologies. Surveillance of the globe is largely provided through MSI sensing such as that given by the Landsat, WorldView, Sentinel etc, however, the practical usefulness of the MSI data set is very limited. This is mainly because of the very limited wave bands in the MSI. One means to remedy this major shortcoming is to extend the MSI into HSI without the need of involving expensive hardware investment. One way to achieve this is through a reconstruction algorithm which turns MSI into HSI using an optimisation approach. The methodology that has been adopted in this thesis is the development of a robust dictionary learning to estimate the endmember (EM) robustly. Once the EM is found the abundance of materials in the scene can be subsequently estimated through a linear unmixing approach.

Although the proposed KMSCD together with the FNNOMP unmixing has shown remarkable results to convert the MSI scene into HSI with a mean 1% DL1NE error (in reflectance), the method relies upon the training of the dictionary through a hyperspectral data set. This is highly undesirable and alternative method, such as the use of comprehensive material database, which is already available, may formulate the next natural step to fulfil the spectral reconstruction without the need of hyperspectral data of the scene for training the dictionary.

Other than that, one should attempt to improve the robustness of the dictionary beyond the SCD-SOMP methodology such that materials irrespective of the abundance distribution in training scene can be learnt independent of errors/statistics in the training

(or material database) set. A possible direction to improve the optimisation is to fuse the SOMP residue within the dictionary update loop.

Finally, it is also seen that the sampling can affect the accuracy and rate of convergence of dictionary learning in the KMSCD algorithm. In particular the KMSCD select one pixel from each class randomly during the training. A possible direction may be to narrow down this randomly sampling by merging the SCD-SOMP and KMSCD through other measures such as similarity check to reduce the number of candidates to a minimum. The ultimate goal is to converge with the minimum number of EM required without compromising the accuracy.

Moreover, existing atmospheric compensation algorithms cannot extract the intrinsic reflectance of the materials when they are subjected to various degree of irradiance. This induces errors in the estimate abundance of materials particularly for those in the shadows. Research on the illumination independent remote sensing has been around for a number of years, however, the progress in this area has been very slow. It is recommended that research effort on the illumination independent scene reconstruction should be in the top priority in order to exploit the full potential of remote sensing for various multidisciplinary applications.

References

[1] Harris geospatial website. https://www.harrisgeospatial.com/Support/Self-Help-Tools/Help-Articles/Help-Articles-Detail/ArtMID/10220/ArticleID/16262/Push-Broom-and-Whisk-Broom-Sensors, (accessed December 25, 2019).

[2] P WT Yuen and M Richardson. An introduction to hyperspectral imaging and its application for security, surveillance and target acquisition. *The Imaging Science Journal*, 58(5):241–253, 2010.

[3] Ayan Chatterjee and Peter W. T. Yuen. Endmember learning with k-means through scd model in hyperspectral scene reconstructions. *Journal of Imaging*, 5(11), 2019.

[4] Jason T. Ward. *Realistic Texture in Simulated Thermal Infrared Imagery*. PhD thesis, Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, 2008.

[5] W. Pereira, S. Richtsmeier, S. Carr, S. Kharabash, and A. Brady. A comparison of mcscene and cameosim simulations of a real scene. In *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4, June 2014.

[6] Roger Evans. Modeling of soc-700 hyperspectral imagery with the cameo-sim code. In *Proceedings of the 2007 Ground Systems Modeling, Validation & Testing Conference*, October 2007.

[7] Eoin O'Keefe Itor James, Mark Richardson. Comparison of empirical and predicted ultraviolet aircraft signatures. *Optical Engineering*, 58(2):1 – 13 – 13, 2019.

[8] Alexander Agathos. *Hyperpectral image acquisition*, 2019 (accessed 25 November 2019).

[9] H. Li, H. Shen, Q. Yuan, H. Zhang, L. Zhang, and L. Zhang. Quality improvement of hyperspectral remote sensing images: A technical overview. In *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5, Aug 2016.

[10] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, April 2012.

[11] A. S. Charles, B. A. Olshausen, and C. J. Rozell. Learning sparse codes for hyperspectral imagery. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):963–978, Sep. 2011.

[12] X. Han, J. Yu, J. Luo, and W. Sun. Reconstruction from multispectral to hyperspectral image using spectral library-based dictionary learning. *IEEE Transactions on Geoscience and Remote Sensing*, 57(3):1325–1335, March 2019.

[13] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006.

[14] X. Fu, W. Ma, T. Chan, and J. M. Bioucas-Dias. Self-dictionary sparse regression for hyperspectral unmixing: Greedy pursuit and pure pixel search are related. *IEEE Journal of Selected Topics in Signal Processing*, 9(6):1128–1141, Sep. 2015.

[15] Jeroen Degerickx, Akpona Okujeni, Marian-Daniel Iordache, Martin Hermy, Sebastian Van der Linden, and Ben Somers. A novel spectral library pruning technique for spectral unmixing of urban land cover. *Remote Sensing*, 9(6), 2017.

[16] X. Fu, W. Ma, J. M. Bioucas-Dias, and T. Chan. Semiblind hyperspectral unmixing in the presence of spectral library mismatches. *IEEE Transactions on Geoscience and Remote Sensing*, 54(9):5171–5184, Sep. 2016.

[17] Alexander W. Houlbrook, Marilyn A. Gilmore, Ian R. Moorhead, David R. Filbee, Colin A. Stroud, G. Hutchings, and Albert Kirk. Scene simulation for camouflage assessment. In *Proc. SPIE 4029, Targets and Backgrounds VI: Characterization, Visualization, and the Detection Process*, volume 4029, 2000.

[18] T. Cooley, G. P. Anderson, G. W. Felde, M. L. Hoke, A. J. Ratkowski, J. H. Chetwynd, J. A. Gardner, S. M. Adler-Golden, M. W. Matthew, A. Berk, L. S. Bernstein, P. K. Acharya, D. Miller, and P. Lewis. Flaash, a modtran4-based atmospheric correction algorithm, its application and validation. In *IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages 1414–1418 vol.3, June 2002.

[19] Stuart O.J. Crichton, Sascha M. Kirchner, Victoria Porley, Stefanie Retz, Gardis von Gersdorff, Oliver Hensel, Martin Weygandt, and Barbara Sturm. Classification of organic beef freshness using vnir hyperspectral imaging. *Meat Science*, 129:20 – 27, 2017.

[20] Susan K. Meerdink, Dar A. Roberts, Keely L. Roth, Jennifer Y. King, Paul D. Gader, and Alexander Koltunov. Classifying california plant species temporally using airborne hyperspectral imagery. *Remote Sensing of Environment*, 232:111308, 2019.

[21] Patrizia Piro, Michele Porti, Simone Veltri, Emanuela Lupo, and Monica Moroni. Hyperspectral monitoring of green roof vegetation health state in sub-mediterranean climate: Preliminary results. *Sensors*, 17(4), 2017.

[22] X. Briottet, Y. Boucher, A. Dimmeler, A. Malaplate, A. Cini, M. Diani, H. Bekman, P. Schwering, T. Skauli, I. Kasen, I. Renhorn, L. Klasén, M. Gilmore, and D. Oxford. Military applications of hyperspectral imagery. In Wendell R. Watkins and Dieter Clement, editors, *Targets and Backgrounds XII: Characterization and Representation*, volume 6239, pages 82 – 89. International Society for Optics and Photonics, SPIE, 2006.

[23] A. Chatterjee and P. W. T. Yuen. Sample selection with somp for robust basis recovery in sparse coding dictionary learning. *IEEE Letters of the Computer Society*, 2(3):28–31, Sep. 2019.

[24] Usman A. Zahidi, Ayan Chatterjee, and Peter W. T. Yuen. A radiative transfer model-based multi-layered regression learning to estimate shadow map in hyperspectral images. *Machine Learning and Knowledge Extraction*, 1(3):904–927, 2019.

[25] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson. Pca-based edge-preserving features for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7140–7151, Dec 2017.

[26] M. D. Farrell and R. M. Mersereau. On the impact of pca dimension reduction for hyperspectral detection of difficult targets. *IEEE Geoscience and Remote Sensing Letters*, 2(2):192–195, April 2005.

[27] Yong Ma, Qiwen Jin, Xiaoguang Mei, Xiaobing Dai, Fan Fan, Hao Li, and Jun Huang. Hyperspectral unmixing with gaussian mixture model and low-rank representation. *Remote Sensing*, 11(8), 2019.

[28] Y. Zhou, A. Rangarajan, and P. D. Gader. A gaussian mixture model representation of endmember variability in hyperspectral unmixing. *IEEE Transactions on Image Processing*, 27(5):2242–2256, May 2018.

[29] D. C. Heinz and Chein-I-Chang. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(3):529–545, March 2001.

[30] Jonas Franke, Dar A. Roberts, Kerry Halligan, and Gunter Menz. Hierarchical multiple endmember spectral mixture analysis (mesma) of hyperspectral imagery for urban environments. *Remote Sensing of Environment*, 113(8):1712 – 1723, 2009.

[31] A. Zare and K. C. Ho. Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing. *IEEE Signal Processing Magazine*, 31(1):95–104, Jan 2014.

[32] C. Chang. Adaptive linear spectral mixture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 55(3):1240–1253, March 2017.

[33] J. B. Greer. Sparse demixing of hyperspectral images. *IEEE Transactions on Image Processing*, 21(1):219–228, Jan 2012.

[34] Y. Liu, Y. Guo, F. Li, L. Xin, and P. Huang. Sparse dictionary learning for blind hyperspectral unmixing. *IEEE Geoscience and Remote Sensing Letters*, 16(4):578–582, April 2019.

[35] J. M. P. Nascimento and J. M. B. Dias. Vertex component analysis: a fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, April 2005.

[36] J. Li, A. Agathos, D. Zaharie, J. M. Bioucas-Dias, A. Plaza, and X. Li. Minimum volume simplex analysis: A fast algorithm for linear hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 53(9):5067–5082, Sep. 2015.

[37] J. Li, J. M. Bioucas-Dias, A. Plaza, and L. Liu. Robust collaborative nonnegative matrix factorization for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6076–6090, Oct 2016.

[38] C. Chang, C. Wu, and H. Chen. Random pixel purity index. *IEEE Geoscience and Remote Sensing Letters*, 7(2):324–328, April 2010.

[39] C. Chang. A review of virtual dimensionality for hyperspectral imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(4):1285–1305, April 2018.

[40] P. Benner, V. Novaković, A. Plaza, E. S. Quintana-Ortí, and A. Remón. Fast and reliable noise estimation for hyperspectral subspace identification. *IEEE Geoscience and Remote Sensing Letters*, 12(6):1199–1203, June 2015.

[41] Michael E. Winter. N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In Michael R. Descour and Sylvia S. Shen, editors, *Imaging Spectrometry V*, volume 3753, pages 266 – 275. International Society for Optics and Photonics, SPIE, 1999.

[42] J. Li and J. M. Bioucas-Dias. Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data. In *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages III – 250–III – 253, July 2008.

[43] Y. E. Salehani, S. Gazor, and M. Cheriet. A new weighted lp-norm for sparse hyperspectral unmixing. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1255–1259, 2017.

[44] S. Zhang, J. Li, H. Li, C. Deng, and A. Plaza. Spectral–spatial weighted sparse regression for hyperspectral image unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3265–3276, June 2018.

[45] J. Li, J. M. Bioucas-Dias, and A. Plaza. Collaborative nonnegative matrix factorization for remotely sensed hyperspectral unmixing. In *2012 IEEE International Geoscience and Remote Sensing Symposium*, pages 3078–3081, July 2012.

[46] S. Matiz and K. E. Barner. Inductive conformal predictor for sparse coding classifiers: Applications to image classification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3307–3311, May 2019.

[47] W. Luo, W. Liu, D. Lian, J. Tang, L. Duan, X. Peng, and S. Gao. Video anomaly detection with sparse coding inspired deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[48] C. Bao, H. Ji, Y. Quan, and Z. Shen. Dictionary learning for sparse coding: Algorithms and convergence analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1356–1369, July 2016.

[49] Y. Chen, G. Li, and Y. Gu. Active orthogonal matching pursuit for sparse subspace clustering. *IEEE Signal Processing Letters*, 25(2):164–168, Feb 2018.

[50] J. Li, M. Li, X. Yao, and H. Wang. An adaptive randomized orthogonal matching pursuit algorithm with sliding window for rolling bearing fault diagnosis. *IEEE Access*, 6:41107–41117, 2018.

[51] S. Zhuang, W. Zhao, R. Wang, Q. Wang, and S. Huang. New measurement algorithm for supraharmonics based on multiple measurement vectors model and orthogonal matching pursuit. *IEEE Transactions on Instrumentation and Measurement*, 68(6):1671–1679, June 2019.

[52] B. Ng and L. Rosenberg. Sparse detection in sea-clutter using orthogonal matching pursuit. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–6, April 2019.

[53] M. S. Asif and J. Romberg. Fast and accurate algorithms for re-weighted ℓ_1-norm minimization. *IEEE Transactions on Signal Processing*, 61(23):5905–5916, Dec 2013.

[54] V. Papyan, J. Sulam, and M. Elad. Working locally thinking globally: Theoretical guarantees for convolutional sparse coding. *IEEE Transactions on Signal Processing*, 65(21):5687–5701, Nov 2017.

[55] K. Schnass. Average performance of orthogonal matching pursuit (omp) for sparse approximation. *IEEE Signal Processing Letters*, 25(12):1865–1869, Dec 2018.

[56] J. Wang, S. Kwon, and B. Shim. Generalized orthogonal matching pursuit. *IEEE Transactions on Signal Processing*, 60(12):6202–6216, Dec 2012.

[57] M. Yaghoobi, D. Wu, and M. E. Davies. Fast non-negative orthogonal matching pursuit. *IEEE Signal Processing Letters*, 22(9):1229–1233, Sep. 2015.

[58] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 40, 01 2008.

[59] Aviris. https://aviris.jpl.nasa.gov, (accessed December 25, 2019).

[60] A. M. Bruckstein, M. Elad, and M. Zibulevsky. On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations. *IEEE Transactions on Information Theory*, 54(11):4813–4820, Nov 2008.

[61] Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995.

[62] L. Rencker, W. Wang, and M. D. Plumbley. Multivariate iterative hard thresholding for sparse decomposition with flexible sparsity patterns. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2156–2160, Aug 2017.

[63] Kjersti Engan, Karl Skretting, and John Håkon Husøy. Family of iterative ls-based dictionary learning algorithms, ils-dla, for sparse signal representation. *Digital Signal Processing*, 17(1):32 – 49, 2007.

[64] Dongsheng Gao, Zhentao Hu, and Renzhen Ye. Self-dictionary regression for hyperspectral image super-resolution. *Remote Sensing*, 10(10), 2018.

[65] L. Fang, S. Li, X. Kang, and J. A. Benediktsson. Spectral–spatial classification of hyperspectral images with a superpixel-based discriminative sparse model. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):4186–4201, Aug 2015.

[66] D. J OLSHAUSEN, B. A; FIELD. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature (London)*, 1996.

[67] Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 – 3325, 1997.

[68] Claes Nelsson, Patrik Hermansson, Thomas Winzell, and Stefan Sjoekvist. Benchmarking and validation of ir signature programs: Sensorvision, cameo-sim and radthermir. In *Swedish Defence Research Agency Linkoeping*, 2005.

[69] Usman A. Zahidi, Peter W. T. Yuen, Jonathan Piper, and Peter S. Godfree. An end-to-end hyperspectral scene simulator with alternate adjacency effect models and its comparison with cameosim. *Remote Sensing*, 12(1), 2019.

[70] A. Berk, P. Conforti, R. Kennett, T. Perkins, F. Hawes, and J. van den Bosch. Modtran® 6: A major upgrade of the modtran® radiative transfer code. In *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4, June 2014.

[71] J. Piper. A new dataset for analysis of hyperspectral target detection performance. In *HSI 2014, Hyperspectral Imaging and Applications Conference, October 2014,Coventry UK.*, October 2014.

[72] Brian Gregor Steven M. Adler-Golden Lawrence S. Bernstein, Xuemin Jin. Quick atmospheric correction code: algorithm description and recent upgrades. *Optical Engineering*, 51(11):1 – 12 – 12, 2012.

[73] Hyspex website. https://www.hyspex.com, (accessed December 25, 2019).

[74] I. S. Reed and X. Yu. Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1760–1770, Oct 1990.

[75] N. M. Nasrabadi. Hyperspectral target detection : An overview of current and future challenges. *IEEE Signal Processing Magazine*, 31(1):34–44, Jan 2014.

[76] Kristen Bretney William F. Basener, Brian Allen. Geometry of statistical target detection. *Journal of Applied Remote Sensing*, 11(1):1 – 22 – 22, 2017.

[77] Bin Cao, Ningfang Liao, Yasheng Li, and Haobo Cheng. Improving reflectance reconstruction from tristimulus values by adaptively combining colorimetric and reflectance similarities. *Optical Engineering*, 56(5):1 – 6, 2017.

[78] K. Wang, H. Wang, Z. Wang, Gu Q., Yin Y., Mao L., and Lu Y. Study of spectral reflectance reconstruction based on regularization matrix r method. *Cluster Computing*, 22:493–502, 2017.

[79] Andre Beckus, Alexandru Tamasan, Aristide Dogariu, Ayman F. Abouraddy, and George K. Atia. On the inverse problem of source reconstruction from coherence measurements. *J. Opt. Soc. Am. A*, 35(6):959–968, Jun 2018.

[80] Shang Zhang, Yuhan Dong, Hongyan Fu, Shao-Lun Huang, and Lin Zhang. A spectral reconstruction algorithm of miniature spectrometer based on sparse optimization and dictionary learning. *Sensors*, 18(2), 2018.

[81] Johan Jacobsson. Terrain model generation for the infra red scene simulation software sensorvision(tm). Master's thesis, School of Computer Science and Engineering, Royal Institute of Technology, Stockholm, Sweden, 2005.

[82] Kefeng Sun, Yu Li, Jiaobo Gao, Jun Wang, Jilong Wang, Junhu Xie, Na Ding, and Dandan Sun. Simulation system of airborne FLIR searcher. In Mircea Guina, Haimei Gong, Zhichuan Niu, and Jin Lu, editors, *International Symposium on Optoelectronic Technology and Application 2014: Infrared Technology and Applications*, volume 9300, pages 480 – 485. International Society for Optics and Photonics, SPIE, 2014.

[83] A. Zuliang, Z. Limin, and Z. Jianting. A practical method of texture segmentation and transformation for radar image simulation. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, volume 3, pages 311–317, May 2012.

[84] Qiong Zhou. Dynamic scene simulation technology used for infrared seeker. In Jeffery Puschell, Hai mei Gong, Yi Cai, Jin Lu, and Jin dong Fei, editors, *International Symposium on Photoelectronic Detection and Imaging 2009: Advances in Infrared Imaging and Applications*, volume 7383, pages 792 – 797. International Society for Optics and Photonics, SPIE, 2009.

[85] S. A. Robila and A. Gershman. Spectral matching accuracy in processing hyperspectral data. In *International Symposium on Signals, Circuits and Systems, 2005. ISSCS 2005.*, volume 1, pages 163–166 Vol. 1, July 2005.

[86] J. D. Shepherd and J. R. Dymond. Correcting satellite imagery for the variance of reflectance and illumination with topography. *International Journal of Remote Sensing*, 24(17):3503–3514, 2003.

[87] Yingzi Du, Chein-I Chang, Hsuan Ren, Chein-Chi Chang, James O. Jensen, and Francis M. D'Amico. New hyperspectral discrimination measure for spectral characterization. *Optical Engineering*, 43(8):1777 – 1786, 2004.

[88] R. R. Nidamanuri and B. Zbell. Normalized spectral similarity score (ns^3) as an efficient spectral library searching method for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(1):226–240, March 2011.

[89] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd Edition*. John Wiley & Sons AND Sons LTD, July 2000.

[90] Ke Wang, Huiqin Wang, Zhan Wang, Qinghua Gu, Ying Yin, Li Mao, and Ying Lu. Study of spectral reflectance reconstruction based on regularization matrix r method. *Cluster Computing*, 22(1):493–502, Jan 2019.

[91] Y. Niu and B. Wang. Hyperspectral target detection using learned dictionary. *IEEE Geoscience and Remote Sensing Letters*, 12(7):1531–1535, July 2015.

[92] M. A. Toksöz, K. G. Toker, and G. G. Güngör. An automatic dictionary construction framework for sparsity-based hyperspectral target detectors. *IEEE Geoscience and Remote Sensing Letters*, 15(8):1264–1268, Aug 2018.

[93] Unique function in matlab. https://uk.mathworks.com/help/matlab/ref/unique.html, (accessed December 25, 2019).

[94] Unique function in python. https://docs.scipy.org/doc/numpy/reference/generated/numpy.unique.html, (accessed December 25, 2019).

[95] Geoffrey M. Smith and Edward J. Milton. The use of the empirical line method to calibrate remotely sensed data to reflectance. *International Journal of Remote Sensing*, 20(13):2653–2662, 1999.

[96] W. M. Baugh and D. P. Groeneveld. Empirical proof of the empirical line. *International Journal of Remote Sensing*, 29(3):665–672, 2008.

Appendix A

Reconstruction accuracy of SCD-SOMP atomic dictionary

This appendix presents the quantified results of comparing two algorithm, precisely the C-SCD and proposed algorithm SCD-SOMP which replaces the random selection in C-SCD algorithm with pixels with the maximum simultaneous orthogonal matching pursuit (SOMP) residue at each training iteration. The SOMP residue estimates the fitting error of all dictionary atoms simultaneously.

The reconstruction of trace materials are measured with Reed-Xiaoli (RX) anomaly detection and Adaptive Cosine Estimator (ACE) target detection with Orange Perspex as target material. The RX is quantified in section A.1 with Pearson correlation coefficient comparing the Ground Truth (GT) RX detection with the reconstruction RX score. And, the ACE detection result is quantified with the area under the curve (AUC) of receiver operating characteristics in section A.2. Towards the end of this appendix, section A.3 presents the DL1NE and MD showing the reconstruction error of the background pixels (dominating materials in the scene like vegetation materials in the Selene scene).

A.1 Anomaly detection with RX

Number of Atoms	C-SCD	SCD-SOMP
10	2.81e-01	8.96e-01
20	4.24e-01	9.09e-01
30	4.44e-01	9.42e-01
40	4.68e-01	9.32e-01
50	7.09e-01	9.28e-01
Mean	4.65e-01	9.22e-01
Std	1.54e-01	1.86e-02

Table A.1: Pearson correlation of RX anomaly detection between the ground truth input image and reconstructed image. Results show a ≈100% mean improvement and ≈8.3x in standard deviation (std) than C-SCD model.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	4.68e-01	5.69e-01	7.09e-01	6.75e-01	5.07e-01	5.86e-01	1.04e-01
SCD-SOMP	9.32e-01	9.32e-01	9.32e-01	9.32e-01	9.26e-01	9.31e-01	3.0e-03

Table A.2: Reproducible run of table A.1 with 40 atoms. Results show a ≈43% mean improvement and ≈73x in standard deviation (std) over C-SCD.

A.2 Target detection with ACE

Number of Atoms	C-SCD	SCD-SOMP
10	6.32e-01	9.98e-01
20	8.16e-01	9.95e-01
30	8.80e-01	9.95e-01
40	8.33e-01	9.95e-01
50	9.99e-01	9.98e-01
Mean	8.33e-01	9.96e-01
Std	1.33e-01	1.7e-03

Table A.3: Area under curve (AUC) for the target detection of Orange Perspex material with ACE with increase in the number of dictionary atoms. Results show a ≈20% mean improvement and ≈78x in standard deviation (std) than C-SCD model.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	8.38e-01	9.24e-01	9.99e-01	9.98e-01	8.98e-01	9.31e-01	6.85e-02
SCD-SOMP	9.95e-01	9.95e-01	9.95e-01	9.95e-01	9.96e-01	9.95e-01	7.42e-04

Table A.4: Area under curve (AUC) for the reproducibility of the detection results of table A.3 with 40 dictionary atoms. Results show a ≈7% mean improvement and ≈92x in standard deviation (std) over C-SCD.

A.3 MD and DL1NE reconstruction error

Manhattan Distance (MD) error

Number of atoms	C-SCD	SCD-SOMP
10	4.57e+0	4.04e+0
20	2.69e+0	3.21e+0
30	2.53e+0	2.87e+0
40	3.93e+0	2.71e+0
50	2.35e+0	2.58e+0
Mean	3.21e+0	3.08e+0
Std	9.77e-01	5.82e-01

Table A.5: Reconstruction error measured in Manhattan distance (MD) with increase in the number of dictionary atoms. Results show a ≈4% improvement in the mean reconstruction error of the proposed SCD-SOMP compared with the C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	3.93e+0	2.43e+0	2.43e+0	2.66e+0	2.57e+0	2.80e+0	6.36e-01
SCD-SOMP	2.71e+0	2.71e+0	2.71e+0	2.71e+0	2.71e+0	2.71e+0	2.01e-03

Table A.6: Mean Manhattan distance (MD) reconstruction error of five runs under the same training parameter. Results show a ≈3.2% improvement in the mean and ≈316x improvement in standard deviation of the proposed SCD-SOMP compared with the C-SCD algorithm.

Differential ℓ_1 norm error (DL1NE)

Number of atoms	C-SCD	SCD-SOMP
10	3.16e-01	3.17e-01
20	2.27e-01	3.45e-01
30	2.33e-01	3.40e-01
40	4.51e-01	3.21e-01
50	2.26e-01	3.16e-01
Mean	2.91e-01	3.28e-01
Std	9.75e-02	1.45e-02

Table A.7: Reconstruction error measured in differential ℓ_1 norm error (DL1NE) with increase in the number of dictionary atoms. Results show that the existing C-SCD algorithm has $\approx$11% better result the mean reconstruction error compared with the proposed SCD-SOMP.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	4.51e-01	2.19e-01	2.14e-01	2.15e-01	2.26e-01	2.65e-01	1.04e-01
SCD-SOMP	3.21e-01	3.21e-01	3.21e-01	3.21e-01	3.31e-01	3.23e-01	4.53e-03

Table A.8: Mean differential ℓ_1 norm error (DL1NE) reconstruction error of five runs under the same training parameter. Results show that the existing C-SCD algorithm has $\approx$18% better result the mean reconstruction error compared with the proposed SCD-SOMP.

Appendix B

Reconstruction accuracy of KMSCD

This appendix presents the quantified results of comparing two algorithm, precisely the C-SCD and proposed algorithm KMSCD which replaces the random selection in C-SCD algorithm with a single pixels from each k-means classification cluster increasing the probability of selecting different materials. Two dictionaries are presented in this appendix, namely atomic dictionaty and endmember (EM) dictionary. In the decomposition of a signal sample 'y' to the dictionary 'D' and its representation 'a' (Y = Da), the atomic dictionary normalises all dictionary atoms (in this case $\ell_2 = 1$) with positive condition on both the 'D' and 'a', and the EM dictionary normalises the representation for sum-to-one abundance ($\ell_1 = 1$) with positive condition on both the 'D' and 'a'. The EM dictionary is a statistical form of spectral unmixing.

The reconstruction of trace materials are measured with Reed-Xiaoli (RX) anomaly detection and Adaptive Cosine Estimator (ACE) target detection with Orange Perspex as target material. The RX is quantified in section B.1 with Pearson correlation coefficient comparing the Ground Truth (GT) RX detection with the reconstruction RX score. And, the ACE detection result is quantified with the area under the curve (AUC) of receiver operating characteristics in section B.2. Towards the end of this appendix, section B.3 presents the DL1NE and MD showing the reconstruction error of the background pixels (dominating materials in the scene like vegetation materials in the Selene scene), and section B.4 presents the error measurements of Virginia City and Paso Robles-Monterey datasets.

B.1 Anomaly detection with RX

B.1.1 Numerical accuracy with atomic dictionary

Number of Atoms	C-SCD	KMSCD
10	2.81e-01	5.15e-01
20	4.24e-01	7.11e-01
30	4.44e-01	8.18e-01
40	4.68e-01	9.41e-01
50	7.09e-01	9.31e-01
Mean	4.65e-01	7.83e-01
Std	1.54e-01	1.77e-01

Table B.1: Pearson correlation of Reed-Xiaoli (RX) anomaly detection between the ground truth input image and reconstructed image with increasing number of atomic dictionary. Results show a ≈69.6% mean improvement over C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	4.68e-01	5.69e-01	7.09e-01	6.75e-01	5.07e-01	5.86e-01	1.04e-01
KMSCD	9.41e-01	9.27e-01	7.60e-01	9.26e-01	7.67e-01	8.64e-01	9.22e-02

Table B.2: Reproducible run of table B.1 with 40 atoms. Results show a ≈47.4% mean improvement and ≈1.1x in standard deviation (std) over C-SCD.

B.1.2 Numerical accuracy with EM dictionary

Number of EMs	C-SCD unmix	KMSCD unmix
10	4.73e-01	6.96e-01
20	4.61e-01	7.83e-01
30	4.58e-01	7.35e-01
40	4.54e-01	7.64e-01
50	4.38e-01	8.72e-01
Mean	4.57e-01	7.70e-01
Std	1.29e-02	6.60e-02

Table B.3: Pearson correlation of Reed-Xiaoli (RX) anomaly detection between the ground truth input image and reconstructed image with increasing number of EM dictionary. Results show a ≈68.5% mean improvement over C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD unmix	4.54e-01	4.29e-01	4.38e-01	4.29e-01	4.67e-01	4.44e-01	1.67e-02
KMSCD unmix	7.64e-01	8.65e-01	7.45e-01	7.07e-01	7.68e-01	7.70e-01	5.87e-02

Table B.4: Reproducible run of table B.3 with 40 atoms. Results show a ≈73.4% mean improvement over C-SCD.

B.2 Target detection with ACE

B.2.1 Numerical accuracy with atomic dictionary

Number of Atoms	C-SCD	KMSCD
10	6.32e-01	9.43e-01
20	8.16e-01	9.98e-01
30	8.80e-01	9.99e-01
40	8.33e-01	9.99e-01
50	9.99e-01	9.99e-01
Mean	8.33e-01	9.99e-01
Std	1.33e-01	2.49e-02

Table B.5: Area under curve (AUC) of target detection of Orange Perspex material with ACE with increase in the number of dictionary atoms. Results show a ≈20% mean improvement and ≈5.3x in standard deviation over C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	8.38e-01	9.24e-01	9.99e-01	9.98e-01	8.98e-01	9.31e-01	6.85e-02
KMSCD	9.99e-01	9.99e-01	9.99e-01	9.99e-01	9.99e-01	9.99e-01	1.24e-04

Table B.6: Area under curve (AUC) for the reproducibility of the detection results of table B.5 with 40 atoms. Results show a ≈7% mean improvement and ≈552.4x in standard deviation (std) over C-SCD.

B.2.2 Numerical accuracy with EM dictionary

Number of EMs	C-SCD unmix	KMSCD unmix
10	7.09e-01	9.97e-01
20	8.39e-01	9.99e-01
30	9.75e-01	9.99e-01
40	7.42e-01	9.99e-01
50	7.76e-01	9.94e-01
Mean	8.29e-01	9.98e-01
Std	1.06e-01	2.10e-02

Table B.7: Area under curve (AUC) for the target detection of Orange Perspex material with ACE with increase in the number of EM dictionary. Results show a ≈20.4% mean improvement and ≈5x in standard deviation over C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD unmix	7.42e-01	6.56e-01	7.47e-01	6.71e-01	7.56e-01	7.15e-01	4.70e-02
KMSCD unmix	9.99e-01	9.99e-01	9.99e-01	9.95e-01	9.99e-01	9.98e-01	1.90e-03

Table B.8: Area under curve (AUC) for the reproducibility of the detection results of table B.7 with 40 atoms. Results show a ≈40% mean improvement and ≈525x in standard deviation (std) over C-SCD.

B.3 MD and DL1NE reconstruction error

B.3.1 Numerical accuracy with atomic dictionary

Manhattan Distance (MD) error

Number of atoms	C-SCD	KMSCD
10	4.57e+0	3.60e+0
20	2.69e+0	3.05e+0
30	2.53e+0	2.6e+0
40	3.93e+0	2.44e+0
50	2.35e+0	2.40e+0
Mean	3.21e+0	2.82e+0
Std	9.77e-01	5.07e-01

Table B.9: Reconstruction error measured in Manhattan distance (MD) with increase in the number of dictionary atoms. Results show a ≈12% improvement in the mean reconstruction error and ≈48% in the standard deviation of the proposed KMSCD compared with the C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	3.93e+0	2.43e+0	2.43e+0	2.66e+0	2.57e+0	2.80e+0	6.36e-01
KMSCD	2.44e+0	2.44e+0	2.43e+0	2.46e+0	2.43e+0	2.44e+0	1.15e-02

Table B.10: Mean Manhattan distance (MD) reconstruction error of five runs under the same training parameter. Results show a ≈13% improvement in the mean and ≈55x improvement in standard deviation of the proposed KMSCD compared with the C-SCD algorithm.

Differential ℓ_1 norm error (DL1NE)

Number of atoms	C-SCD	KMSCD
10	3.16e-01	4.27e-01
20	2.27e-01	2.09e-01
30	2.33e-01	2.28e-01
40	4.51e-01	2.05e-01
50	2.26e-01	1.96e-01
Mean	2.91e-01	2.53e-01
Std	9.75e-02	9.78e-02

Table B.11: Reconstruction error measured in differential ℓ_1 norm error (DL1NE) with increase in the number of dictionary atoms. Results show that an improvement of ≈13% in the mean DL1NE of the proposed KMSCD compared with the C-SCD algorithm.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	4.51e-01	2.19e-01	2.14e-01	2.15e-01	2.26e-01	2.65e-01	1.04e-01
KMSCD	2.05e-01	1.89e-01	2.06e-01	2.12e-01	1.79e-01	1.98e-01	1.37e-02

Table B.12: Mean differential ℓ_1 norm error (DL1NE) reconstruction error of five runs under the same training parameter. Results show an improvement of ≈25.3% in the mean DL1NE and ≈7.6x improvement in the standard deviation of the proposed KMSCD compared with the C-SCD algorithm.

B.3.2 Numerical accuracy with EM dictionary

Manhattan Distance (MD) error

Number of EMs	C-SCD unmix	KMSCD unmix
10	2.73e+0	2.77e+0
20	2.37e+0	2.26e+0
30	2.31e+0	2.27e+0
40	2.41e+0	2.19e+0
50	2.38e+0	2.19e+0
Mean	2.44e+0	2.34e+0
Std	1.69e-01	2.45e-02

Table B.13: Reconstruction error measured in Manhattan distance (MD) with increase in the number of endmembers. Results show a ≈4.1% improvement in the mean reconstruction error and ≈6.9x in standard deviation of the proposed KMSCD compared with the C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD unmix	2.41e+0	2.36e+0	2.37e+0	2.38e+0	2.38e+0	2.38e+0	1.60e-01
KMSCD unmix	2.19e+0	2.22e+0	2.23e+0	2.27e+0	2.22e+0	2.23e+0	2.86e-02

Table B.14: Mean Manhattan distance (MD) reconstruction error of five runs under the same training parameter. Results show a ≈6.7% improvement in the mean error and ≈5.6x in standard deviation of the proposed KMSCD compared with the C-SCD algorithm.

Differential ℓ_1 norm error (DL1NE)

Number of EMs	C-SCD unmix	KMSCD unmix
10	2.94e-01	1.58e-01
20	1.92e-01	1.46e-01
30	3.25e-01	2.48e-01
40	4.01e-01	2.13e-01
50	4.04e-01	2.42e-01
Mean	3.23e-01	2.02e-01
Std	8.76e-02	4.73e-02

Table B.15: Mean differential ℓ_1 norm error (DL1NE) reconstruction error with increase in the number of EM dictionary. Results show a ≈37% mean improvement and ≈1.85x in standard deviation over C-SCD unmixing.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD unmix	4.01e-01	2.46e-01	2.97e-01	3.42e-01	3.19e-01	3.21e-01	5.73e-02
KMSCD unmix	2.13e-01	2.40e-01	2.24e-01	2.83e-01	2.42e-01	2.41e-01	2.65e-02

Table B.16: Mean differential ℓ_1 norm error (DL1NE) reconstruction error of five runs under the same training parameter with 40 endmembers. Results show a ≈25% mean improvement and ≈2.16x in standard deviation over C-SCD unmixing.

B.4 Numerical accuracy with atomic dictionary on datasets other than Selene

B.4.1 Virginia City Dataset

Manhattan Distance (MD) error

	0817-1220 test scene		0817-1259 test scene	
Number of atoms	C-SCD	KMSCD	C-SCD	KMSCD
10	1.00e+0	6.96e-01	1.00e+0	6.58e-01
20	5.07e-01	4.81e-01	4.81e-01	4.85e-01
30	4.35e-01	3.69e-01	4.28e-01	3.90e-01
40	3.95e-01	3.39e-01	3.93e-01	3.69e-01
50	3.58e-01	3.02e-01	3.56e-01	3.37e-01
Mean	5.40e-01	4.37e-01	5.32e-01	4.48e-01
Std	2.66e-01	1.59e-01	2.68e-01	1.30e-01

Table B.17: Reconstruction error measured in Manhattan distance (MD) with increase in the number of dictionary atoms. Results show a ≈19 to 24% improvement in the mean reconstruction error and ≈67 to 106% in the standard deviation of the proposed KMSCD compared with the C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	3.95e-01	3.88e-01	3.96e-01	3.94e-01	3.94e-01	3.94e-01	3.20e-02
KMSCD	3.39e-01	3.83e-01	3.85e-01	3.86e-01	3.76e-01	3.73e-01	1.99e-02

(a) 0817-1220 test scene

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	3.93e-01	3.97e-01	3.98e-01	3.99e-01	3.91e-01	3.96e-01	3.80e-02
KMSCD	3.69e-01	3.98e-01	4.08e-01	4.09e-01	3.97e-01	3.96e-01	1.64e-02

(b) 0817-1259 test scene

Table B.18: Mean Manhattan distance (MD) reconstruction error of five runs under the same training parameter. Results show a ≈0 to 6% improvement in the mean and ≈61 to 132% improvement in standard deviation of the proposed KMSCD compared with the C-SCD algorithm.

Differential ℓ_1 norm error (DL1NE)

	0817-1220 test scene		0817-1259 test scene	
Number of atoms	**C-SCD**	**KMSCD**	**C-SCD**	**KMSCD**
10	2.58e-01	1.96e-01	2.73e-01	1.92e-01
20	1.31e-01	9.31e-02	1.18e-01	8.61e-02
30	1.17e-01	9.42e-02	1.10e-01	1.04e-01
40	9.58e-02	7.49e-02	8.75e-02	7.65e-02
50	9.53e-02	6.20e-02	8.50e-02	7.46e-02
Mean	1.40e-01	1.04e-01	1.35e-01	1.07e-01
Std	6.80e-02	5.33e-02	7.85e-02	4.91e-02

Table B.19: Reconstruction error measured in differential ℓ_1 norm error (DL1NE) with increase in the number of dictionary atoms. Results show a ≈26 to 35% improvement in the mean reconstruction error and ≈28 to 60% in the standard deviation of the proposed KMSCD compared with the C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	9.58e-02	9.48e-02	9.00e-02	8.91e-02	9.12e-02	9.22e-02	3.00e-02
KMSCD	7.49e-02	1.07e-01	9.05e-02	9.68e-02	8.93e-02	9.17e-02	1.17e-02

(a) 0817-1220 test scene

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	8.75e-02	8.91e-02	8.20e-02	8.34e-02	8.29e-02	8.50e-02	3.10e-02
KMSCD	7.65e-02	7.83e-02	8.73e-02	9.36e-02	8.69e-02	8.45e-02	7.10e-03

(b) 0817-1259 test scene

Table B.20: Mean differential ℓ_1 norm error (DL1NE) reconstruction error of five runs under the same training parameter. Results show a ≈0.6% improvement in the mean and ≈156 to 337% improvement in standard deviation of the proposed KMSCD compared with the C-SCD algorithm.

B.4.2 Paso Robles-Monterey Dataset

Manhattan Distance (MD) error

Number of atoms	C-SCD	KMSCD
10	20.28e+0	12.12e+0
20	6.39e+0	3.58e+0
30	3.50e+0	3.35e+0
40	3.06e+0	2.74e+0
50	2.60e+0	2.49e+0
Mean	7.17e+0	4.86e+0
Std	7.48e+0	4.08e+0

Table B.21: Reconstruction error measured in Manhattan distance (MD) with increase in the number of dictionary atoms. Results show a ≈32% improvement in the mean reconstruction error and ≈45% in the standard deviation of the proposed KMSCD compared with the C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	3.06e+0	2.88e+0	2.85e+0	2.75e+0	2.95e+0	2.90e+0	1.17e-01
KMSCD	2.74e+0	2.64e+0	2.72e+0	2.72e+0	2.84e+0	2.73e+0	7.40e-02

Table B.22: Mean Manhattan distance (MD) reconstruction error of five runs under the same training parameter. Results show a ≈6% improvement in the mean and ≈37% improvement in standard deviation of the proposed KMSCD compared with the C-SCD algorithm.

Differential ℓ_1 norm error (DL1NE)

Number of atoms	C-SCD	KMSCD
10	11.81e+0	1.81e+0
20	1.03e+0	6.74e-01
30	7.83e-01	6.98e-01
40	6.84e-01	6.18e-01
50	6.74e-01	5.46e-01
Mean	2.30e+0	8.68e-01
Std	4.93e+0	5.27e-01

Table B.23: Reconstruction error measured in differential ℓ_1 norm error (DL1NE) with increase in the number of dictionary atoms. Results show a $\approx$62% improvement in the mean reconstruction error and $\approx$89% in the standard deviation of the proposed KMSCD compared with the C-SCD.

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	Std
C-SCD	6.84e-01	7.03e-01	8.03e-01	6.66e-01	6.79e-01	7.07e-01	5.50e-02
KMSCD	6.18e-01	6.40e-01	6.65e-01	6.57e-01	6.68e-01	6.50e-01	2.08e-02

Table B.24: Mean differential ℓ_1 norm error (DL1NE) reconstruction error of five runs under the same training parameter. Results show a $\approx$8% improvement in the mean and $\approx$39% improvement in standard deviation of the proposed KMSCD compared with the C-SCD algorithm.

Appendix C

MATLAB Source Codes

C.1 Dictionary Learning Functions

C.1.1 SCD-SOMP dictionary learning

```
%% SCD (Sparse Coding Dictionary Learning) with SOMP sampling
% Published paper (IEEE LOCS): https://doi.org/10.1109/LOCS.2019.2938446
% SCD-SOMP Code: https://codeocean.com/capsule/5120060/
% SCD source code originally authored by Adam Charles and this code is
% modified by Ayan Chatterjee. Email: ayan@outlook.com
% This code supports compatible GPU (optional). Please see relevant links.

%% Relevant Links (Last accessed 10 September 2019)
% Adam's SCD Paper (IEEE JSTP): https://doi.org/10.1109/JSTSP.2011.2149497
% Adam's SCD code: http://adamsc.mycpanel.princeton.edu/documents/
    Dictionary_Learning_Library_v1-0.zip
% SD-SOMP paper (IEEE JSTP): https://doi.org/10.1109/JSTSP.2015.2410763
% SD-SOMP source code: https://mx.nthu.edu.tw/~tsunghan/download/
    Demo_LqSDSOMP.rar

%% Inputs
% Image -> 3D hyperspectral or multispectral cube
% num_atoms -> number of atoms to learn
% maxIter -> maximum number of iterations

function dictionary = SCD_TrainSOMP(Image, num_atoms, maxIter)

clc;
[h, w, nB] = size(Image);
Image = reshape(single(Image), [h*w, nB]);
```

```
24
25 opts.save_name = 'temp_1.mat';       % Save Name in case of errors
26 opts.sparse_type = 'l1ls_nneg';      % Choose to use l1ls for sparsification
       OMP_qr
27 opts.grad_type = 'norm';             % Choose weather to include the Forb
     norm in E(a,D)
28 opts.n_elem = num_atoms;             % Number of dictionary elements
29 opts.iters = maxIter;                % Number of learning iterations
30 opts.in_iter = 200;                  % Number of internal iterations
31 opts.GD_iters = 1;                   % Basis Gradient Descent iterations
32 opts.step_size = 0.01;               % Initial Step Size for Gradient
     Descent
33 opts.decay = 0.9998;                 % Step size decay factor
34 opts.lambda = 0.5;                   % Lambda Value for Sparsity
35 opts.tol = 0.001;                    % Sparsification Tolerance
36 opts.verb = 1;                       % Default to no verbose output
37 opts.ssim_flag = 0;                  % Default to no normalization between
     samples
38 opts.std_min = 0.1;                  % Default to min. sample standard
     deviation of 0.1
39
40 dictionary_initial = abs(rand(nB, opts.n_elem)); %create a random initial
     dictionary
41 dictionary_initial = dictionary_initial./(ones(nB, 1)*sqrt(sum(
     dictionary_initial.^2, 1))); % Basis normalized for l2 norm
42 dictionary = learn_dictionary(Image', dictionary_initial,
     @l1ls_nneg_wrapper, opts); %learn dictionary
43 dictionary = dictionary';
44
45 end
46
47 function [dictionary_end] = learn_dictionary(data_obj, initial_dict,
     infer_handle, opts)
48 % OPTIONS: Make sure that the correct options are set and that all
49 % necessary variables are available or defaulted.
50 if ~isfield(opts, 'ssim_flag')
51     opts.ssim_flag = 0; % Default to no normalization between samples
52 end
53
54 if (~isfield(opts, 'std_min'))
55     opts.std_min = 0.1; % Default to min. sample standard deviation of 0.1
56     warning('Inputs:UnspecifiedParam',  ...
57         ['Min sample STD not set by user!! Using STDmin = 0.1. ', ...
58         'This is probably bad! ', ...
59         'Ctrl-C and restarting with a specified value is recommended.'])
```

```
60 end
61
62 if ~isfield(opts, 'save_name')
63     date_str = date;
64     opts.save_name = [date_str(8:end), date_str(3:7), date_str(1:2), ...
65         'Dictionary_' num2str(opts.n_elem), 'Elems_', num2str(opts.lambda),
       ...
66         'lambda.mat'];
67     fprintf('Save name not specified, saving as %s...\n', opts.save_name)
68 end
69 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
70 %% Error Checking
71 if opts.n_elem ~= size(initial_dict, 2)
72     error('Dimension mismatch between opts.n_elem and initial dictionary
      size!')
73 end
74
75 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
76 %% Initializations and Dimention Extraction
77 % Initialize Basis
78 dictionary_n = initial_dict;
79 % Iteration counter initialization
80 iter_num = 0;
81 % Initialize step size
82 step_s = opts.step_size;
83 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
84 %% Run Algorithm
85 fprintf('Educating your basis...\n')
86 basic_cell.options = opts;
87 while iter_num < opts.iters
88     try
89         %% Get Training Data.
90         if(gpuDeviceCount >= 1)
91             SOMP_residue = estSOMPresidue(gpuArray(dictionary_n), gpuArray(
      data_obj));
92             SOMP_residue = gather(SOMP_residue);
93         else
94             SOMP_residue = estSOMPresidue(dictionary_n, data_obj);
95         end
96         [~, data_use_ind] = sort(SOMP_residue, 'descend'); % descending
      order of SOMP
97         data_use_ind = data_use_ind(1:opts.in_iter);
98         x_im = data_obj(:, data_use_ind); % Initialize matricies that will
      be populated during the actual learning
99         %% Interior loop: find sparse coefficients
```

```
100         coef_vals = gen_multi_infer(dictionary_n, x_im, infer_handle, opts)
    ;
101         %% Minimize the energy w.r.t. the dictionary using gradient descent
102         dictionary_n = dictionary_update(x_im, dictionary_n, coef_vals,
    step_s, opts);
103         iter_num = iter_num + 1; % update the iteration count
104         if opts.verb == 1
105             %Spit out info
106             im_snr = mean(sum(x_im.^2, 1)./sum((x_im - dictionary_n*
    coef_vals).^2, 1));
107             disp(strcat("Iter: ", num2str(iter_num),", SNR: ", num2str(
    im_snr),", mean SOMP: ", num2str(mean(SOMP_residue)),", step size is ",
     num2str(step_s)));
108         end
109         % Update the step size
110         step_s = step_s*opts.decay;
111     catch ME
112         fprintf('Saving last dictionary before error...\n')
113         basic_cell.dictionary = dictionary_n;
114         basic_cell.iter = iter_num;
115         eval(sprintf('save %s basic_cell;', opts.save_name));
116         fprintf(ME.message)
117         fprintf('The program failed. Your dictionary at the last iteration
    was saved.')
118         rethrow(ME)
119     end
120 end
121 dictionary_end = dictionary_n;
122
123 end
124
125 % Function added by Ayan
126 function SOMP_residue = estSOMPresidue(dictionary_n, data_obj)
127 P = eye(size(dictionary_n, 1)) - dictionary_n*pinv(dictionary_n);
128 SOMP_residue = sum(abs(P*data_obj).^2, 1).^0.5; % l-2 SOMP
129 end
130
131 function coef_vals = gen_multi_infer(dictionary_n, x_im, infer_hand, opts)
132 % Initialize coefficients
133 coef_vals = zeros(opts.n_elem, opts.in_iter);
134 %% Perform the L1-regulized optimization on the data
135 for index_in = 1:opts.in_iter
136     coef_vals(:, index_in) = feval(infer_hand, dictionary_n, x_im(:,
    index_in), opts);
137 end
```

```
138
139 end
140
141 function coef_vals = l1ls_nneg_wrapper(dictionary_n, x_im, opts)
142 coef_vals = l1_ls_nonneg(dictionary_n, x_im, opts.lambda, opts.tol, 1);
143 end
144
145 function x = l1_ls_nonneg(dictionary, varargin)
146 % IPM PARAMETERS
147 MU              = 2;        % updating parameter of t
148 MAX_NT_ITER     = 400;      % maximum IPM (Newton) iteration
149
150 % LINE SEARCH PARAMETERS
151 ALPHA           = 0.01;     % minimum fraction of decrease in the objective
152 BETA            = 0.5;      % stepsize decrease factor
153 MAX_LS_ITER     = 100;      % maximum backtracking line search iteration
154
155 if (nargin >= 3)
156     [~, nBands] = size(dictionary);
157     y  = varargin{1};
158     lambda = varargin{2};
159     varargin = varargin(3:end);
160 else
161     x = [];
162     return;
163 end
164
165 % VARIABLE ARGUMENT HANDLING
166 t0          = min(max(1,1/lambda),nBands/1e-3);
167 defaults    = {1e-3,false,1e-3,5000,ones(nBands,1),t0};
168 given_args = ~cellfun('isempty',varargin);
169 defaults(given_args) = varargin(given_args);
170 [reltol,~,eta,pcgmaxi,x,t] = deal(defaults{:});
171
172 f = -x;
173
174 % RESULT/HISTORY VARIABLES
175 pobjs = [] ; dobjs = [] ; sts = [] ; pitrs = []; pflgs = [];
176 dobj  =-Inf;
177 s   = Inf;
178 conv_iter_n  = 0;
179 conv_flag  = 0 ;
180
181 ntiter  = 0;
182 lsiter  = 0;
```

```
183 conv_vector =  zeros(nBands, 1);
184
185 % diagxtx = diag(At*A);
186 diagxtx = 2*ones(nBands,1);
187 %-----------------------------------------------------------
188 %                MAIN LOOP
189 %-----------------------------------------------------------
190
191 for ntiter = 0:MAX_NT_ITER
192
193     z = dictionary*x-y;
194
195     %-----------------------------------------------------------
196     %        CALCULATE DUALITY GAP
197     %-----------------------------------------------------------
198
199     nu = 2*z;
200
201     minAnu = min(dictionary'*nu);
202     if (minAnu < -lambda)
203         nu = nu*lambda/(-minAnu);
204     end
205     pobj   =  z'*z+lambda*sum(x,1);
206     dobj   =  max(-0.25*nu'*nu-nu'*y,dobj);
207     gap    =  pobj - dobj;
208
209     pobjs = [pobjs pobj];
210     dobjs = [dobjs dobj];
211     sts = [sts s];
212     pflgs = [pflgs conv_flag];
213     pitrs = [pitrs conv_iter_n];
214
215     %-----------------------------------------------------------
216     %   STOPPING CRITERION
217     %-----------------------------------------------------------
218
219     if (gap/abs(dobj) < reltol)
220         return; %if solved
221     end
222     %-----------------------------------------------------------
223     %        UPDATE t
224     %-----------------------------------------------------------
225     if (s >= 0.5)
226         t = max(min(nBands*MU/gap, MU*t), t);
227     end
```

```
228
229     %-------------------------------------------------------------
230     %       CALCULATE NEWTON STEP
231     %-------------------------------------------------------------
232
233     d1 = (1/t)./(x.^2);
234
235     % calculate gradient
236     gradphi = dictionary' * (z*2) + lambda-(1/t)./x;
237
238     % calculate vectors to be used in the preconditioner
239     prb     = diagxtx + d1;
240
241     % set pcg tolerance (relative)
242     normg   = norm(gradphi);
243     pcgtol  = min(1e-1,eta*gap/min(1,normg));
244
245     if (ntiter ~= 0 && conv_iter_n == 0)
246         pcgtol = pcgtol*0.1;
247     end
248
249     [conv_vector, conv_flag, ~, conv_iter_n,~] =...
250         pcg(@AXfunc_l1_ls, -gradphi, pcgtol, pcgmaxi,@Mfunc_l1_ls,[],...
251         conv_vector, dictionary, d1, 1./prb); %the 4 inputs into both
    functions
252
253     if (conv_flag == 1)
254         conv_iter_n = pcgmaxi;
255     end
256
257     %-------------------------------------------------------------
258     %   BACKTRACKING LINE SEARCH
259     %-------------------------------------------------------------
260     phi = z'*z+lambda*sum(x)-sum(log(-f))/t;
261     s = 1.0;
262     gdx = gradphi'*conv_vector;
263     for lsiter = 1:MAX_LS_ITER
264         newx = x+s*conv_vector;
265         newf = -newx;
266         if (max(newf) < 0)
267             newz    =  dictionary*newx-y;
268             newphi  =  newz'*newz+lambda*sum(newx)-sum(log(-newf))/t;
269             if (newphi-phi <= ALPHA*s*gdx)
270                 break;
271             end
```

```
272         end
273         s = BETA*s;
274     end
275     if (lsiter == MAX_LS_ITER)
276         break;
277     end % exit by BLS
278 
279     x = newx;
280     f = newf;
281 end
282 
283 %-----------------------------------------------------------
284 %       ABNORMAL TERMINATION (FALL THROUGH)
285 %-----------------------------------------------------------
286 return;
287 
288 end
289 
290 %-----------------------------------------------------------
291 %       COMPUTE AX (PCG)
292 %-----------------------------------------------------------
293 function y = AXfunc_l1_ls(conv_vector, dict, d1, ~)
294 y = (dict'*((dict*conv_vector)*2)) + d1.*conv_vector;
295 end
296 
297 %-----------------------------------------------------------
298 %       COMPUTE P^{-1}X (PCG)
299 %-----------------------------------------------------------
300 function y = Mfunc_l1_ls(conv_vector, ~, ~, prb_inv)
301 y = prb_inv.*conv_vector;
302 end
303 
304 function dict_new = dictionary_update(x_im, dict_old, coef_vals, step_s,
    opts)
305 
306 % function dict_new = dictionary_update(x_im, dictionary_old, coef_vals,
307 % step_s, opts)
308 %
309 % Takes a gradient step with respect to the sparsity inducing energy
310 % function.
311 %
312 % Inputs:
313 %   x_im        - Data samples over which to average the gradient step
314 %   dict_old    - The previous dictionary (used to infer the coefficients)
315 %   coef_vals   - The inferred coefficients for x_im using dict_old
```

```
%    step_s       - The step size to take in the gradient direction
%    opts         - Options for the particular problem (outlined in
%                   learn_dictionary.m)
%
% Outputs:
%    dict_new     - The new dictionary after the gradient step
%
% Last Modified 6/4/2010 - Adam Charles

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Take a gradient step
if strcmp(opts.grad_type, 'norm')
    for index2 = 1:opts.GD_iters
        % Take a step in the negative gradient of the basis:
        % Minimizing the energy:
        % E = ||x-Da||_2^2 + lambda*||a||_1^2
        % Update The basis matrix
        updateTerm = (x_im - dict_old*coef_vals)*coef_vals';
        dict_new = dict_old + step_s*updateTerm;
        dict_new = dict_new*diag(1./(sqrt(sum(dict_new.^2)))); % Re-
    normalize the basis for l2 norm
    end
elseif strcmp(opts.grad_type, 'forb')
    for index2 = 1:opts.GD_iters
        % Take a step in the negative gradient of the basis:
        % This time the Forbenious norm is used to reduce unused
        % basis elements. The energy function being minimized is
        % then:
        % E = ||x-Da||_2^2 + lambda*||a||_1^2 + ||D||_F^2

        % Update The basis matrix
        dict_new = dict_old + (step_s)*((x_im - dict_old*coef_vals)*
    coef_vals'...
            - opts.lambda2*2*dict_old)*diag(1./(1+sum(coef_vals ~= 0, 2)));
    end
end
end
```

C.1.2 Rapid OMP Represenntation

```
%% Rapid Orthogonal Matching Pursuit Estimate
% Authored by Ayan Chatterjee (ayan@outlook.com) on 23 December 2019

function sparseRep = RapidOMP(signals, basis, maxBasisLim, maxErrorTol)

```

```
%% pre-process
[~, numSignals] = size(signals);
[~, numBasis] = size(basis);
basisMatch = single(zeros(numBasis, numSignals));
sparseRep = single(zeros(numBasis, numSignals));
residue = signals;
if(~exist('maxErrorTol', 'var'))
    maxErrorTol = 1e-05; % change as required
end
signalsRemaining = 1:numSignals;

%% starting rapid OMP algorithm
for lim = 1:maxBasisLim
    %% basis selection
    basisProject = abs(basis' * residue);
    for basisCount = 1:numBasis
        selectedBasisPos = sum(basisProject > basisProject(basisCount, :)) == 0;
        basisMatch(basisCount, signalsRemaining(selectedBasisPos == 1)) = 1; % these samples will get allocated #basisCount basis
    end

    %% estimate residue
    [uniqueBasisMatch, ~, invBasisPos] = unique(basisMatch', 'rows');
    for invIter = 1:size(uniqueBasisMatch, 1)
        matchPos = find(uniqueBasisMatch(invIter, :) == 1);
        invBasis = basis(:, matchPos);
        invBasis = pinv(invBasis' * invBasis) * invBasis';
        sparseRep(matchPos, invBasisPos == invIter) = invBasis * signals(:, invBasisPos == invIter);
    end
    residue = signals - basis * sparseRep; % residue (r) = y - Da
    signalsRemaining = find(sum(abs(residue)) > maxErrorTol);
    residue = residue(:, signalsRemaining);
end

end
```

C.1.3 K-SVD with Rapid OMP

```
%% K-SVD dictionary learning algorithm with rapid OMP for representation
% Authored by Ayan Chatterjee (ayan@outlook.com) on 24 December 2019. K-SVD algorithm was published with IEEE TSP journal (doi:10.1109/TSP.2006.881199).

```

```
function [dictionary, sparseRep, t] = KSVD(Image, numAtoms, maxIter,
    maxAtomPerPixel)

%% pre-process
[h, w, b] = size(Image);
Image = reshape(single(Image), [h*w, b])'; % single datatype to save space
dictionary = single(randn(b, numAtoms));
dictionary = dictionary ./ sum(dictionary.^2).^0.5; % normalise the initial
     basis
t = zeros(1, maxIter);

%% run K-SVD algorithm
for iter = 1:maxIter
    tic;
    sparseRep = RapidOMP(Image, dictionary, maxAtomPerPixel);
    residue = Image - dictionary * sparseRep;
    for atomPos = 1:numAtoms
        ImageSubsetPos = find(sparseRep(atomPos, :) > 0);
        if(numel(ImageSubsetPos) == 0)
            continue; % this atom is not contributing to any pixel
        end
        ImageSubset = residue(:, ImageSubsetPos) + dictionary(:, atomPos)*
    sparseRep(atomPos, ImageSubsetPos);
        [U, S, V] = svds(double(ImageSubset), 1, 'L'); % svds accepts
    double datatype
        dictionary(:, atomPos) = single(U); % U is already normalized
        sparseRep(atomPos, ImageSubsetPos) = S*V';
        residue(:, ImageSubsetPos) = ImageSubset - dictionary(:, atomPos)*
    sparseRep(atomPos, ImageSubsetPos);
    end
    t(iter) = toc; % captures time taken per training iteration
    disp(strcat("Iter: ", num2str(iter), ", mean residue: ", num2str(mean(
    sum(abs(residue)))), ", time taken: ", num2str(t(iter))));
end

sparseRep = reshape(sparseRep', [h, w, numAtoms]);

end
```

C.1.4 ILSDLA with Rapid OMP

```
%% Iterative least squares (ILS) dictionary learning algorithm (DLA) with
    rapid OMP for representation
% Authored by Ayan Chatterjee (ayan@outlook.com) on 1 January 2020.

```

```
function [dictionary, sparseRep, t] = ILSDLA(Image, numAtoms, maxIter,
    maxAtomPerPixel)

%% pre-process
[h, w, b] = size(Image);
Image = reshape(single(Image), [h*w, b])'; % single datatype to save space
dictionary = single(randn(b, numAtoms));
dictionary = dictionary ./ sum(dictionary.^2).^0.5; % normalise the initial
     basis
t = zeros(1, maxIter);

% run the ILS-DLA
for iter = 1:maxIter
    tic;
    sparseRep = RapidOMP(Image, dictionary, maxAtomPerPixel);
    dictionary = Image*sparseRep'*pinv(sparseRep*sparseRep');
    dictionary = dictionary ./ sum(dictionary.^2).^0.5; % normalize the
    dictionary
    residue = Image - dictionary * sparseRep;
    t(iter) = toc; % captures time taken per training iteration
    disp(strcat("Iter: ", num2str(iter), ", mean residue: ", num2str(mean(
    sum(abs(residue)))), ", time taken: ", num2str(t(iter))));
end

sparseRep = reshape(sparseRep', [h, w, numAtoms]);

end
```

C.2 KMSCD+FNNOMP for scene simulators

C.2.1 Function to train KMSCD endmembers

```
%% KMSCD Endmember Learning with FNNOMP for representation
% C-SCD source code originally authored by Adam Charles and this code is
    modified by Ayan Chatterjee. Email: ayan@outlook.com. Please see
    relevant links.
% Note: Replace FNNOMP with Non-negative least squares function for KMSCD
    unmixing. FNNOMP abundance is recommended for hyperspectral scene
    simulator constraints.

%% Relevant Links (Last accessed 29 September 2019)
% C-SCD Paper (IEEE JSTP): https://doi.org/10.1109/JSTSP.2011.2149497
% C-SCD code: http://adamsc.mycpanel.princeton.edu/documents/
    Dictionary_Learning_Library_v1-0.zip
```

```
8
9 %% Inputs
10 % Image -> 3D hyperspectral or multispectral cube
11 % num_endmembers -> number of atoms to learn
12 % maxMatsPerPixel - > maximum number of endmember materials per pixel
13 % maxIter -> maximum number of iterations
14
15 function endmembers = KMSCDUnmix(Image, num_endmembers, maxMatsPerPixel,
       maxIter)
16
17 clc;
18 [h, w, nB] = size(Image);
19 Image = reshape(single(Image), [h*w, nB]);
20
21 opts.save_name = 'temp_1.mat';      % Save Name in case of errors
22 opts.grad_type = 'norm';            % Choose weather to include the Forb
       norm in E(a,D)
23 opts.n_elem = num_endmembers;       % Number of endmember materials
24 opts.max_sparsity = maxMatsPerPixel - 1; % maximum number of endmember
       materials per pixel. One endmember reserved for illumination changes.
25 opts.iters = maxIter;               % Number of learning iterations
26 opts.in_iter = 200;                 % Number of internal iterations
27 opts.GD_iters = 1;                  % Basis Gradient Descent iterations
28 opts.step_size = 0.01;              % Initial Step Size for Gradient
       Descent
29 opts.decay = 0.9998;                % Step size decay factor
30 opts.lambda = 0.5;                  % Lambda Value for Sparsity
31 opts.verb = 1;                      % Default to no verbose output
32
33 dictionary_initial = abs(rand(nB, opts.n_elem)); %create a random initial
       dictionary
34 dictionary_initial = dictionary_initial./(ones(nB, 1)*sqrt(sum(
       dictionary_initial.^2, 1))); % Basis normalized for l2 norm
35 kclass = kmeans(Image, opts.in_iter); % unsupervised clustering with kmeans
36 endmembers = train_endmembers(Image', dictionary_initial, kclass, opts); %
       learn dictionary
37 endmembers = endmembers';
38
39 end
40
41 function endmembers_end = train_endmembers(data_obj, initial_dict, kclass,
       opts)
42 % OPTIONS: Make sure that the correct options are set and that all
43 % necessary variables are available or defaulted.
44
```

```
if ~isfield(opts, 'save_name')
    date_str = date;
    opts.save_name = [date_str(8:end), date_str(3:7), date_str(1:2), ...
        'Dictionary_' num2str(opts.n_elem), 'Elems_', num2str(opts.lambda), ...
        'lambda.mat'];
    fprintf('Save name not specified, saving as %s...\n', opts.save_name)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Error Checking
if opts.n_elem ~= size(initial_dict, 2)
    error('Dimension mismatch between opts.n_elem and initial dictionary size!')
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Initializations and Dimention Extraction
% Initialize Basis
endmembers_n = initial_dict;
% Iteration counter initialization
iter_num = 0;
% Initialize step size
step_s = opts.step_size;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Run Algorithm
fprintf('Educating your basis...\n')
basic_cell.options = opts;
while iter_num < opts.iters
    try
        %% Get Training Data.
        data_use_ind = zeros(1, opts.in_iter);
        parfor i = 1:opts.in_iter
            matpos = find(kclass == i);
            data_use_ind(i) = matpos(ceil(numel(matpos)*rand(1)));
        end
        x_im = data_obj(:, data_use_ind); % Initialize matricies that will be populated during the actual learning
        %% Interior loop: find sparse coefficients
        coef_vals = gen_multi_infer(double(endmembers_n), double(x_im), opts);
        %% Minimize the energy w.r.t. the dictionary using gradient descent
        endmembers_n = endmembers_update(x_im, endmembers_n, coef_vals, step_s, opts);
        iter_num = iter_num + 1; % update the iteration count
        if opts.verb == 1
```

```
85              %Spit out info
86              im_snr = mean(sum(x_im.^2, 1)./sum((x_im - endmembers_n*
    coef_vals).^2, 1));
87              disp(strcat("Iter: ", num2str(iter_num),", SNR: ", num2str(
    im_snr),", step size is ", num2str(step_s)));
88          end
89          % Update the step size
90          step_s = step_s*opts.decay;
91      catch ME
92          fprintf('Saving last dictionary before error...\n')
93          basic_cell.dictionary = endmembers_n;
94          basic_cell.iter = iter_num;
95          eval(sprintf('save %s basic_cell;', opts.save_name));
96          fprintf(ME.message)
97          fprintf('The program failed. Your dictionary at the last iteration
    was saved.')
98          rethrow(ME)
99      end
100 end
101 endmembers_end = endmembers_n;
102
103 end
104
105 function coef_vals = gen_multi_infer(endmembers_n, x_im, opts)
106 % Initialize coefficients
107 coef_vals = zeros(opts.n_elem, opts.in_iter);
108 for index_in = 1:opts.in_iter
109     coef_vals(:, index_in) = fnnomp([endmembers_n; ones(1, opts.n_elem)], [
    x_im(:,index_in); 1], opts.max_sparsity, 0);
110 end
111 end
112
113 function endmembers_new = endmembers_update(x_im, endmembers_old, coef_vals
    , step_s, opts)
114
115 % function endmembers_new = endmembers_update(x_im, dictionary_old,
    coef_vals,
116 % step_s, opts)
117 %
118 % Takes a gradient step with respect to the sparsity inducing energy
119 % function.
120 %
121 % Inputs:
122 %   x_im        - Data samples over which to average the gradient step
123 %   endmembers_old    - The previous dictionary (used to infer the
```

```
    coefficients)
%    coef_vals    - The inferred coefficients for x_im using endmembers_old
%    step_s       - The step size to take in the gradient direction
%    opts         - Options for the particular problem (outlined in
%                   learn_dictionary.m)
%
% Outputs:
%    endmembers_new     - The new dictionary after the gradient step
%
% Last Modified 6/4/2010 - Adam Charles

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Take a gradient step
if strcmp(opts.grad_type, 'norm')
    for index2 = 1:opts.GD_iters
        % Take a step in the negative gradient of the basis:
        % Minimizing the energy:
        % E = ||x-Da||_2^2 + lambda*||a||_1^2
        % Update The basis matrix
        updateTerm = (x_im - endmembers_old*coef_vals)*coef_vals';
        endmembers_new = endmembers_old + step_s*updateTerm;
        % endmembers_new(endmembers_new < 0) = 0; % endmember reflectance
    cannot be less than 0
    end
elseif strcmp(opts.grad_type, 'forb')
    for index2 = 1:opts.GD_iters
        % Take a step in the negative gradient of the basis:
        % This time the Forbenious norm is used to reduce unused
        % basis elements. The energy function being minimized is
        % then:
        % E = ||x-Da||_2^2 + lambda*||a||_1^2 + ||D||_F^2

        % Update The basis matrix
        endmembers_new = endmembers_old + (step_s)*((x_im - endmembers_old*
    coef_vals)*coef_vals'...
            - opts.lambda2*2*endmembers_old)*diag(1./(1+sum(coef_vals ~= 0,
     2)));
    end
end
end
```

C.2.2 Function to estimate FNNOMP abundance for the test scene

```
function [endmembers, abundance] = estFNNOMPabundance(endmembers, image,
    maxMatsPerPixel)
```

```
clc;
[h, w, b] = size(image);
image = reshape(double(image), [h*w, b]);
image = image';
[nD, ~] = size(endmembers);
endmembers = double(endmembers');

abundance = single(zeros(h*w, nD));
for px = 1:h*w
    abundance(px, :) = single(fnnomp([endmembers; ones(1, nD)], [image(:, px); 1], maxMatsPerPixel-1, 0));
end
m = max(abundance(:));
abundance = abundance / m;
endmembers = endmembers * m;

s = find(sum(abundance) == 0);
endmembers(:, s) = []; % remove endmemembers with zero abundance because they do not contribute to the reconstructed scene.
abundance(:, s) = [];
endmembers = [endmembers, zeros(size(endmembers, 1), 1)]; % add a zeros vector for illumination changes
abundance = [abundance, zeros(size(abundance, 1), 1)];
abundance(:, end) = 1 - sum(abundance(:, 1:end-1), 2);
abundance = reshape(single(abundance), [h, w, size(endmembers, 2)]);
endmembers = endmembers';

end
```

C.2.3 FNNOMP code

```
%%  Fast Non-Negative Orthogonal Matching Pursuit (FNNOMP)
% This code from the original author is uploaded in URL http://www.mehrdadya.com/code/NNOMPv1.0.tar.gz
% FNNOMP Paper (IEEE SPL): https://doi.org/10.1109/LSP.2015.2393637

function x = fnnomp(A, y, maxAtom, tol)

r = y;
[m,n] = size(A);
x = zeros(n,1);
k = 1;
mag = 1;
s = [];
bpr = A'*r;
```

```
Q = zeros(m,maxAtom);
R = zeros(maxAtom);
Rm1 = [];
xs = [];
r_pre = zeros(size(r));

while k <= maxAtom && mag>0 && abs(norm(r_pre,2) - norm(r,2)) >= tol
    done = 0;
    zc = 0;
    Inc = [];
    l = 1;
        bpr(s) = 0;
        [amp,In] = sort(bpr,'descend');
    while ~done
        if amp(l) > 0
            sint = [s;In(l)];
            anew = A(:,sint(end));

            qP=Q(:,1:k-1)'*anew;
            q=anew-Q(:,1:k-1)*(qP);
            nq=norm(q);
            q=q/nq;
            zin = q'*y;

            %%% Positivity Guarantee

            v = qP;
            mu = nq;
            gamma = -Rm1*v/mu;

            xsp = xs;
            if ~isempty(gamma)
            if ~isempty(find(gamma<0, 1))
                vt = abs((xsp./gamma).*(gamma<0));
                zt = min(vt(vt>0));
            else
                zt = inf;
            end
            else
                zt = inf;
            end
            if (zin <= zt)
                if (zin <= zc)
                    s = [s;In(l)];
                    anew = A(:,In(l));
```

```
                    qP=Q(:,1:k-1)'*anew;
                    q=anew-Q(:,1:k-1)*(qP);
                    nq=norm(q);
                    q=q/nq;
                else
                    s = [s;In(l)];
                end
                done = 1;
            else
                if zc>=zin
                    s = [s;In(lc)];
                    anew = A(:,In(lc));
                    qP=Q(:,1:k-1)'*anew;
                    q=anew-Q(:,1:k-1)*(qP);
                    nq=norm(q);
                    q=q/nq;
                    done = 1;
                elseif zt > zc,
                    zc = zt;
                    lc = l;
                    l = l+1;
                else
                    l = l+1;
                end
            end
        else
            done = 1;
            mag = amp(l);
        end
        if done && (mag > 0)
%               R(1:k-1,k)=qP; % Updatin R
%               R(k,k)=nq; % Updatin R
             Q(:,k)=q;
             Rm1 = [Rm1,gamma;zeros(1,size(Rm1,2)),1/mu];
             z(k)=q'*y;
             r_pre = r;
             r = r- q*z(k);
             xs = Rm1*z(1:k)';
        end
    end
    bpr = A'*r;
    k = k+1;
end
xs(xs < 0) = 0;
x(s) = xs;
```

```
end
```

C.3 Create CTX file for CameoSim

```
function EMtoCTXfile(Endmembers, EMlambda, input_image, image_lambda,
    unique_name, sparse_rep)

clc;
maxMats = 4; % maximum limit allowed by cameosim
[h, w, b] = size(input_image);
input_image = reshape(input_image, [h*w, b])';
if(exist('sparse_rep', 'var'))
    sparse_rep = reshape(sparse_rep, [h*w, size(sparse_rep, 3)])';
end

EMcopy = Endmembers; % copy for subset to the input image lambda
if(EMlambda ~= image_lambda)
    EMcopy = spline(EMlambda, Endmembers, image_lambda);
end

if(~exist('sparse_rep', 'var'))
    [sparse_rep, delEMs] = estFNNOMPabundance(EMcopy(:, 1:end-1),
    input_image, maxMats); % zeros vector is the last endmember
    if(delEMs(end)) == size(Endmembers, 2))
        Endmembers(:, delEMs(1:end-1)) = [];
    else
        Endmembers(:, delEMs) = [];
    end
end

materialAllocationMap = zeros(h*w, 2*maxMats);
WB = waitbar(0,'Step 1 of 2: Matching Materials');
for mat = 1:h*w
    CS_abundance = sparse_rep(:, mat);
    [~, nearest_distance_position] = sort(CS_abundance, 'descend');
    nearest_distance_position = nearest_distance_position(1:maxMats);
    abundance = CS_abundance(nearest_distance_position);
    nearest_distance_position = nearest_distance_position(:);
    abundance = abundance(:);
    materialAllocationMap(mat, :) = [nearest_distance_position; abundance];
end
close(WB);

materialAllocationMap = reshape(materialAllocationMap, [h, w, 2*maxMats]);
```

```
[EMfinal, pixel_x, pixel_y, emIndex] = updateMaterialLibrary(Endmembers,
    materialAllocationMap); % cleanup for cameosim
outputImage = zeros(h, w, numel(EMlambda));
abund = emIndex(:, 2:2:size(emIndex, 2));
emIndex_Pixel = emIndex(:, 1:2:size(emIndex, 2));

WB = waitbar(0, 'Step 2 of 2: Reconstructing scene');
for mat = 1:h*w
    outputImage(pixel_y(mat) + 1, pixel_x(mat) + 1, :) = (abund(mat, :) *
    EMfinal(emIndex_Pixel(mat, :), :))/100;
end
close(WB);
outputImage = single(outputImage);
emIndex(:, 1:2:size(emIndex, 2)) = emIndex(:, 1:2:size(emIndex, 2)) - 1;
C = char(zeros(h*w,1));
C(C == 0) = 'C';
materialMap = table;
materialMap.C = C;
materialMap.x_position = pixel_x;
materialMap.y_position = pixel_y;
materialMap.Material_Abundance_Pair = emIndex;
RGB_image = makeRGBimage(outputImage, EMlambda);
CTX_choice = int8(input('\nEnter 1 if you wish to create materials for
    CameoSim..\n'));
if(CTX_choice == 1)
    if(~exist('unique_name', 'var'))
        unique_name = input('\nEnter your unique folder name (without space
    )\n', 's');
    end
    CreateCSMaterialFile(unique_name, RGB_image, EMfinal, EMlambda,
    materialMap);
end

end

function rgbimage = makeRGBimage(imageCube, lambda)
[h, w, ~] = size(imageCube);
rgbimage = zeros(h, w, 3);
lambdaPos = findBandPositions(lambda);
for i = 1:3
  rgbimage(:, :, i) = imageCube(:, :, lambdaPos(i));
    rgbimage(:,:,i) = rgbimage(:,:,i) - min(min(rgbimage(:,:,i)));
    rgbimage(:,:,i) = 3.6 * rgbimage(:,:,i) / max(max(rgbimage(:,:,i)));
end
figure; imagesc(rgbimage); axis image;
```

```
79 end
80
81 function [EMout, pixel_x, pixel_y, emIndex] = updateMaterialLibrary(EMs,
       materialAllocationMap)
82
83 [h, w, ~] = size(materialAllocationMap);
84 EM_material_index = zeros(size(EMs, 1), 1);
85 pixel_x = zeros(h*w, 1);
86 pixel_y = zeros(h*w, 1);
87 emIndex = zeros(h*w, 2 * floor(numel(materialAllocationMap(1, 1, :))/2));
88 count = 0;
89 pixelCount = 0;
90 WB = waitbar(0, 'Updating Matched Materials');
91 for i = 1:h
92     for j = 1:w
93         pixelCount = pixelCount + 1;
94         pixel_x(pixelCount) = j-1;
95         pixel_y(pixelCount) = i-1;
96         pixel_data = squeeze(materialAllocationMap(i, j, :));
97         emPos = pixel_data(1:floor(numel(pixel_data)/2)); %end member
       positions in extisting file
98         checkMaterials = EM_material_index(emPos) == 0;
99         emIndex_Pixel = zeros(floor(numel(pixel_data)/2), 1);
100        for k = 1:floor(numel(pixel_data)/2)
101            if(checkMaterials(k) == 1) %to add new material to file
102                count = count + 1;
103                EM_material_index(pixel_data(k)) = count;
104                EMout(count, :) = EMs(pixel_data(k), :);
105                emIndex_Pixel(k) = count;
106            else % material is already selected
107                emIndex_Pixel(k) = EM_material_index(pixel_data(k));
108            end
109        end
110        emIndex(pixelCount, 1:2:2*floor(numel(pixel_data)/2)) =
       emIndex_Pixel;
111        emAbund = pixel_data(floor(numel(pixel_data)/2)+1:2*floor(numel(
       pixel_data)/2));
112        %start cameoSIM specifications
113        emAbund = uint8(emAbund*100); %convert abundancy to percent as per
       CameoSIM specifications
114        pos100 = sum(emAbund) ~= 100;
115        if(pos100 == 1)
116            maxPos100 = find(emAbund == max(emAbund));
117            diffPos100 = 100 - sum(emAbund);
118            emAbund(maxPos100(1)) = emAbund(maxPos100(1)) + diffPos100;
```

```
119         end
120         %end cameoSIM specifications
121         emIndex(pixelCount, 2:2:2*floor(numel(pixel_data)/2)) = emAbund;
122     end
123     waitbar(i/h);
124 end
125 close(WB);
126 
127 end
128 
129 function CreateCSMaterialFile(unique_name, RGB_image, EndMembers, lambda,
    materialMap)
130 clc;
131 material_header = 'HASH {\n    "ABSORPTIVITY" = FUNC {\n    0.4 0,\n    },\
    n    "DESC" = "",\n    "LAYERS" = UNDEF,\n    "MAJOR" = 0,\n    "MINOR"
     = 0,\n    "OPAQUE" = 1,\n    "REFRACTIVE_INDEX" = FUNC {\n    0.4 1.6,\
    n    },\n\n    "SURFACE_BACK" = UNDEF,\n    "SURFACE_FRONT" = HASH {\n
    "BIDIRECTIONAL_REFLECTANCE" = LIST {\n  },\n    "CLOUD_ALTITUDE" = 2,\n
     "CLOUD_SKYSHINE" = 100,\n    "CLOUD_SUNSHINE" = 100,\n    "COATED" = 0,\
    n "FRESNEL_FRACTION" = 1,\n    "GRAZING_REFLECTIVITY" = 0.8,\n "
    LIMB_DARK_WIDTH" = 2,\n    "MEAN_FACET_SLOPE" = 0.1,\n "SHININESS_EXP"
    = 100,\n    "SPECTRAL_EMISSIVITY" = FUNC {\n        0.4 0.2,\n  },\n
     "SPECTRAL_REFLECTIVITY" = FUNC {\n';
132 material_footer = '},\n "SPECULAR_LOBE_WIDTH" = 0.1,\n  "
    SPECULAR_REFLECTIVITY" = FUNC {\n      0.4 0.5,\n  },\n    "SURFACE_ASH
    " = 0,\n    "SURFACE_BRDF" = 0,\n    "SURFACE_CLOUD" = 0,\n  "
    SURFACE_COOK" = 0,\n    "SURFACE_DIFFUSE" = 1,\n    "SURFACE_SANDFORD" =
     0,\n    },\n    "TEMPERATURE_HISTORY" = FUNC {\n    0    288,\n    },\n
       "TEMP_EXTERIOR" = 1,\n    "TEMP_INSULATED" = 0,\n    "TEMP_ROCK" =
    0,\n    "TEMP_TABULATED" = 0,\n    "THICKNESS" = 5,\n    "
    TRANSLUCENCY_TABLE" = FUNC {\n    0.4 0.5,\n    },\n    "TRANSLUCENT" =
    0,\n    "TRANSMISSIVE" = 0,\n}\n';
133 if ~exist(unique_name, 'dir')
134     mkdir(unique_name);
135 end
136 imwrite(RGB_image, strcat(unique_name, '/', unique_name,'_image.png'));
137 
138 % printing the header file
139 fileID = fopen(strcat(unique_name, '/', unique_name, '_ctx_file.txt'),'w');
140 fprintf(fileID, '# CameoSim image classification data\n\n');
141 fprintf(fileID, strcat('I /opt/insys/share/cameosim/materials/',
    unique_name,'_image.png\n\n'));
142 WB = waitbar(0, 'CTX File Create: Writing endmembers..');
143 for emCount = 1:size(EndMembers, 1)
144     fprintf(fileID, strcat('M', '\t', unique_name, '_endmember_', int2str(
```

```
    emCount), '_mat\n'));
145     %write the endmember file
146     fileID1 = fopen(strcat(unique_name, '/', unique_name, '_endmember_',
    int2str(emCount), '_mat'),'w');
147     fprintf(fileID1, material_header);
148     fclose(fileID1);
149     emData = NaN(3, numel(lambda));
150     emData(1:2, :) = [lambda; EndMembers(emCount, :)];
151     dlmwrite(strcat(unique_name, '/', unique_name, '_endmember_', int2str(
    emCount), '_mat'), emData', '-append', 'delimiter','\t');
152     fileID1 = fopen(strcat(unique_name, '/', unique_name, '_endmember_',
    int2str(emCount), '_mat'),'r+');
153     fileText=fread(fileID1,'*char')';
154     fileText = regexprep(fileText,'NaN',',');
155     fclose(fileID1);
156     fileID1 = fopen(strcat(unique_name, '/', unique_name, '_endmember_',
    int2str(emCount), '_mat'),'w+');
157     fprintf(fileID1,'%s',fileText);
158     fprintf(fileID1, material_footer);
159     fclose(fileID1);
160     waitbar(emCount/size(EndMembers, 1));
161 end
162 close(WB);
163 fprintf(fileID, '\nN 4\n');
164 WB = waitbar(0, 'CTX File Create: Writing pixel information..');
165 for pxCount = 1:size(materialMap, 1) % for all pixels in the scene
166     x_pos = materialMap.x_position(pxCount); y_pos = materialMap.y_position
    (pxCount);
167     Mat_Abund_Pair = materialMap.Material_Abundance_Pair(pxCount,:); %
    extract pixel information
168     abundValue = Mat_Abund_Pair(2:2:end); % abundance index
169     matIndex = Mat_Abund_Pair(1:2:end);
170     matIndex(abundValue == 0) = []; abundValue(abundValue == 0) = [];
171     Mat_Abund_Pair = zeros(1, numel(abundValue)*2);
172     Mat_Abund_Pair(2:2:end) = abundValue;
173     Mat_Abund_Pair(1:2:end) = matIndex;
174     fprintf(fileID, strcat('C', '\t', num2str(x_pos), '\t', num2str(y_pos),
     '\t', num2str(Mat_Abund_Pair), '\n'));
175     waitbar(pxCount/size(materialMap, 1));
176 end
177 close(WB);
178 fclose(fileID);
179
180 end
181
```

```
182 function lambdaPos = findBandPositions(varargin)
183
184 if(nargin == 0)
185     lambdaPos = 1;
186     return;
187 elseif(nargin == 1)
188     lambda_GT = cell2mat(varargin);
189     lambda_find = [0.6329, 0.5510, 0.4528]; %RGB wavelengths
190 elseif(nargin == 2)
191     lambda_GT = varargin{1};
192     lambda_find = varargin{2};
193 end
194
195 checkDomain = find(lambda_GT>min(lambda_find)&lambda_GT<max(lambda_find));
196 if(numel(checkDomain) == 0)
197     lambdaPos = 1;
198     warning('\nInput wavelengths are out of range\n');
199 else
200     lambdaPos = NaN(numel(lambda_find), 1);
201     for i = 1:numel(lambda_find)
202         lambdaDiff = abs(lambda_GT - lambda_find(i));
203         lambdaMin = find(lambdaDiff == min(lambdaDiff));
204         lambdaPos(i) = lambdaMin(1);
205     end
206 end
207
208 end
```